

Aula 06 (Prof. Thiago Cavalcanti)

*IFCE (Prof. Ciência da
Computação-Metodologia e Técnicas da
Comput) Conhecimentos -
2021(Pós-Edital)*

Autor:

**Diego Carvalho, Equipe
Informática e TI, Evandro Dalla
Vecchia Pereira , Pedro Henrique
Chagas Freitas, Thiago Rodrigues**

Implementação de SGBDs relacionais.	3
<i>Objetivos da Aula.....</i>	<i>3</i>
Conceitos básicos	3
<i>O catálogo de dados Relacional.....</i>	<i>6</i>
<i>Tabelas.....</i>	<i>8</i>
<i>Visões</i>	<i>13</i>
<i>Índices</i>	<i>16</i>
<i>Chaves e Relacionamentos.....</i>	<i>18</i>
Restrições de Integridade	22
<i>Mapa mental – Restrições de Integridade.....</i>	<i>25</i>
Regras de Codd	26
<i>Mapa mental – Regras de Codd.....</i>	<i>29</i>
Álgebra relacional	30
<i>Seleção</i>	<i>32</i>
<i>Projeção</i>	<i>35</i>
<i>União.....</i>	<i>36</i>
<i>Intersecção.....</i>	<i>37</i>
<i>Diferença.....</i>	<i>37</i>
<i>Produto Cartesiano.....</i>	<i>38</i>
<i>Junção</i>	<i>39</i>
<i>Divisão.....</i>	<i>41</i>
<i>Tabela com o resumo das operações da Álgebra Relacional.</i>	<i>43</i>
<i>Mapa Mental – Álgebra Relacional</i>	<i>44</i>
Normalização	45
<i>Introdução.....</i>	<i>45</i>
<i>Conceitos Básicos.....</i>	<i>46</i>
<i>Dependência funcional</i>	<i>47</i>
<i>Chave, superchave, chave primária e chave candidata</i>	<i>49</i>
<i>Primeira forma normal (1FN).....</i>	<i>50</i>
<i>Segunda forma normal (2FN).....</i>	<i>51</i>



<i>Terceira forma normal (3FN)</i>	53
<i>Forma normal de Boyce-Codd (FNBC)</i>	56
Questões Comentadas CESPE	58
<i>Questões comentadas</i>	58
<i>Exercícios</i>	70
<i>Gabarito</i>	77
Considerações Finais	78



THIAGO CAVALCANTI
PROFESSOR



IMPLEMENTAÇÃO DE SGBDs RELACIONAIS.

Fala pessoal! Muito bom dia, boa tarde ou boa noite! Vamos começar agora essa aula para conhecermos um pouco mais sobre banco de dados. Nosso objetivo é apresentar os conceitos mais importantes dos assuntos, sistematizar o conhecimento e garantir que você tenha condições de acertar qualquer questão deste tema que venha aparecer na sua prova. Ousado!? Um pouco, talvez. Mas confiamos no nosso trabalho e na sua capacidade de aprendizado. Vamos em frente!

OBJETIVOS DA AULA

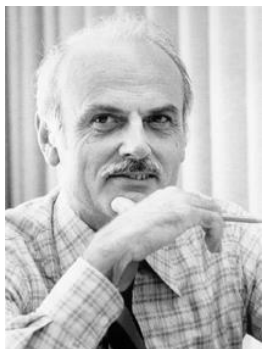
- Conhecer a terminologia de banco de dados relacional.
- Entender o significado das regras de integridade para bancos de dados relacionais.
- Entender o impacto das linhas referenciadas na manutenção de bancos de dados relacionais.
- Entender o significado de cada operador da álgebra relacional.

CONCEITOS BÁSICOS

Já sabemos que um banco de dados é uma **coleção de dados persistentes** que podem ser **compartilhados** e **estar inter-relacionados**. Ou seja, os dados são armazenados de modo permanente, como em um disco magnético. Ele também pode ter múltiplas aplicações e usuários, fazendo acesso aos dados armazenados. Tais dados são vistos como unidades separadas e podem ser ligados para fornecer um quadro completo.

Os sistemas de banco de dados relacionais foram desenvolvidos originalmente por causa da **familiaridade** e **simplicidade**. Como as tabelas são utilizadas para comunicar ideias em vários campos do conhecimento, a terminologia de tabelas, linhas e colunas é facilmente compreendida por diferentes usuários. No modelo, **tanto os dados quanto os relacionamentos são representados em tabelas ou relações**.

Além da familiaridade e simplicidade dos bancos de dados relacionais, há também **uma sólida base matemática**. A matemática dos bancos de dados relacionais implica a conceitualização de tabelas como **conjuntos**. O modelo tem uma base formal sólida, que está estruturada na **teoria dos conjuntos e na lógica de primeira ordem**.



A combinação de familiaridade e simplicidade com a base matemática é tão poderosa, que os SGBDs relacionais são **líderes no mercado de armazenamento de dados operacionais**.

Edgar Frank Codd (o rapaz fumando um charuto cubano na figura ao lado) foi um matemático britânico que desenvolveu o modelo de banco de dados relacional quando era pesquisador da IBM em San

José – Califórnia. É interessante que a IBM não quis implementar as ideias de Codd inicialmente. A empresa tinha uma linha produtos, em especial o IMS/DB, e não queria perder faturamento. O IMS/DB era um SGBD que não utilizava o modelo relacional como fundamento teórico. A IBM tinha um grande volume de receitas advindas deste produto. Assim, não queria perder espaço neste mercado, mesmo que fosse para ela mesma. Mas o que Codd propôs que seria tão revolucionário?

Para responder a essa pergunta, vamos iniciar com uma definição informal do modelo. O modelo relacional é constantemente descrito como tendo os três aspectos a seguir:



Aspecto estrutural: os dados no banco de dados são percebidos pelo usuário como tabelas, e nada além de tabelas.

Aspecto de integridade: essas tabelas satisfazem a certas restrições de integridade ou limitações nos valores, que podem aparecer em cada registro.

Aspecto manipulador: os operadores disponíveis para que o usuário possa manipular essas tabelas – por exemplo, para propósitos de busca de dados – são operadores que derivam tabelas a partir de outras tabelas. Desses operadores, três particularmente importantes são os operadores de restrição, projeção e junção.

Ok! Vamos sair do texto puro e observar uma figura. Ela vai nos ajudar a entender esses conceitos de forma bem mais tranquila. Veja a figura abaixo:



Tabela Professor (**aspecto estrutural**)

Identidade	Nome	Concurso	Comida Favorita
0000001	Eduardo (Dudu)	ICMS-SP	Pudim
0000002	Rosenval Junior	IBAMA	Churrasco
0000003	Ricardo Vale	MIDC	Salada
0000004	Paulo Bilynskyj	Delegado (DHPP)	McDonald's

Colunas:
Identidade (numérica)
Comida favorita (texto)
(**aspecto de integridade**)

Operação:
Projeção com as colunas
Nome e Comida Favorita
(**aspecto manipulador**)

Nome	Comida Favorita
Eduardo (Dudu)	Pudim
Rosenval Junior	Churrasco
Ricardo Vale	Salada
Paulo Bilynskyj	McDonald's

Figura 1 - Aspecto estrutural, de integridade e manipulador do modelo



O modelo relacional é um modelo de dado **representativo** (ou de **implementação**). Como um modelo, ele procura abstrair e organizar os dados de um banco de dados. Para isso, o modelo relacional define um conjunto de conceitos para representação dos dados. O elemento básico deste modelo está definido pelo conceito **de relação**. Um banco de dados seria, portanto, um conjunto de relações. Cada relação pode ser vista como **uma tabela**. Assim, os SGBDs relacionais devem representar os dados sob a forma de **tabelas bidimensionais** organizadas em linhas e colunas.

Observe que os sistemas relacionais só exigem que o banco de dados seja percebido pelo usuário como tabelas. As tabelas são a **estrutura lógica** em um sistema relacional, **não a estrutura física**. No nível físico, os dados podem ser organizados em outros tipos de estrutura – usando arquivos sequenciais, indexação, *hashing*, cadeias de ponteiros etc. – desde que ele **possa mapear essa representação armazenada como tabelas no nível lógico**. Isso é um reflexo da abstração provida pelo modelo relacional.

O termo "estrutura lógica" pretende englobar os níveis conceitual e externo da arquitetura ANSI/SPARC. O detalhe é que os níveis conceitual e externo em um **sistema relacional** serão representados usando o modelo relacional. Os bancos de dados relacionais satisfazem a um princípio muito interessante, chamado **Princípio da Informação**:



Todo o conteúdo de informação do banco de dados é representado de um e somente um modo, ou seja, como valores explícitos em posições de colunas em linhas de tabelas.

Veja que os valores são armazenados em uma tabela. Não existe o conceito de ponteiros conectando uma tabela a outra. Mas o que seriam ponteiros? Tenho certeza de que você fez essa pergunta mentalmente! Ponteiro é um tipo especial de dados que armazena um endereço de memória. Por exemplo, se um ponteiro p armazena o endereço de uma variável i , podemos dizer p aponta para i ou p é o endereço de i . Enfim, reforçando, no modelo relacional, **não existe o conceito de ponteiros conectando uma tabela a outra**.

Endereço na memória	Variável na memória
1000	Ponteiro 1003
1001	1
1002	1
1003	7
1004	2

Ainda não está satisfeito com a explicação acima? Talvez ela seja, de fato, um pouco abstrata. Mas vamos olhar, mais uma vez, para uma imagem e tentar esclarecer, de uma vez por todas, essa característica. Olhe para o modelo abaixo e perceba que temos duas tabelas: empregado (EMP) e departamento (DEPTO). Perceba que a coluna DEPTO# da tabela EMP possui os valores dos departamentos.



DEPTO	DEPTO#	NOMEDEPTO	ORÇAMENTO
	D1	Marketing	10M
	D2	Desenvolvimento	12M
	D3	Pesquisa	5M

EMP	EMP#	NOMEEMP	DEPTO#	SALÁRIO
	E1	Lopez	D1	40K
	E2	Cheng	D1	42K
	E3	Finzi	D2	30K
	E4	Saito	D2	35K

Sendo assim, existe uma conexão entre a linha D1 da tabela DEPTO e a linha E1 da tabela EMP. Essa conexão é representada pelo **aparecimento do valor D1** na tabela EMP. Vamos agora resumir os conceitos que vimos até aqui.

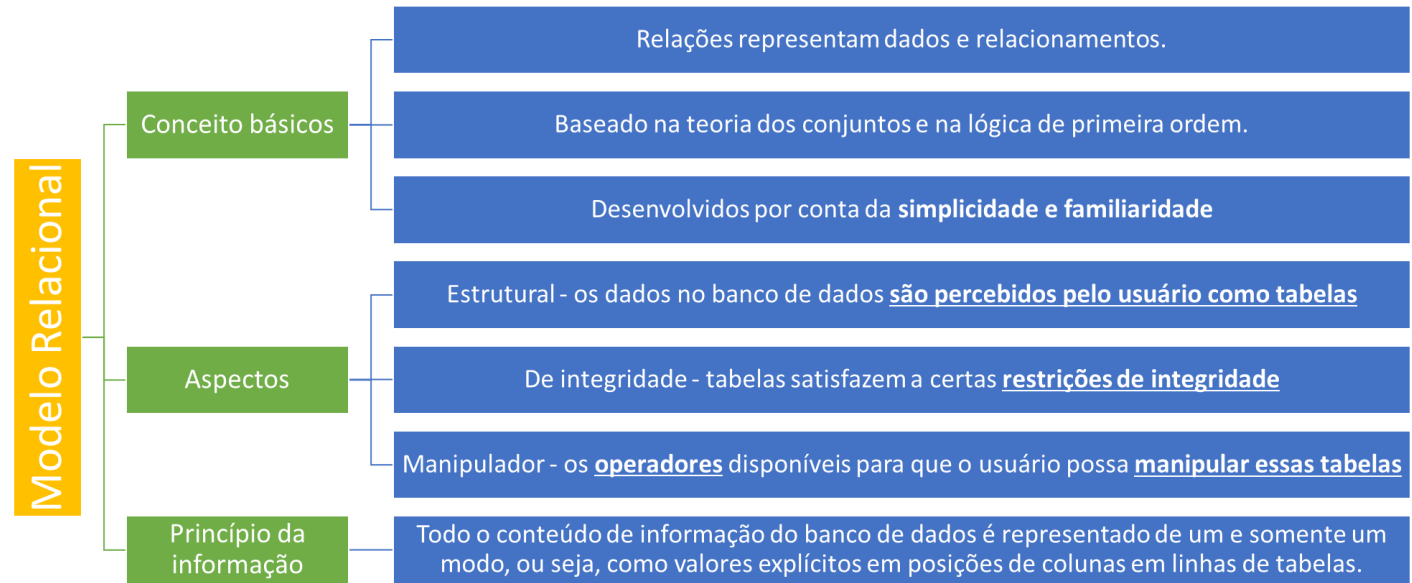


Figura 2 - Conceitos básicos do modelo relacional

O CATÁLOGO DE DADOS RELACIONAL

O Sistema de Gerenciamento de Banco de Dados precisa fornecer uma **função de catálogo ou dicionário**. O catálogo é o lugar em que – dentre outras coisas – todos os diversos esquemas (externo, conceitual, interno) e todos os mapeamentos correspondentes (externo/conceitual, conceitual/interno) são mantidos.

Em outras palavras, o catálogo contém informações detalhadas. Às vezes, elas são chamadas informações do descritor ou metadados, com objetos *Re/Vars*, índices, usuários, restrições de integridade, restrições de segurança, e assim por diante.

As informações do descritor são essenciais para que o sistema faça seu trabalho de modo apropriado. Por exemplo, o otimizador utiliza informações do catálogo a respeito de índices e outras estruturas físicas de armazenamento, bem como muitas outras informações, para ajudá-lo a decidir como implementar as requisições do usuário. Da mesma forma, o subsistema de autorização utiliza informações do catálogo sobre usuários e restrições de segurança para conceder ou negar tais requisições.



ESQUEMATIZANDO

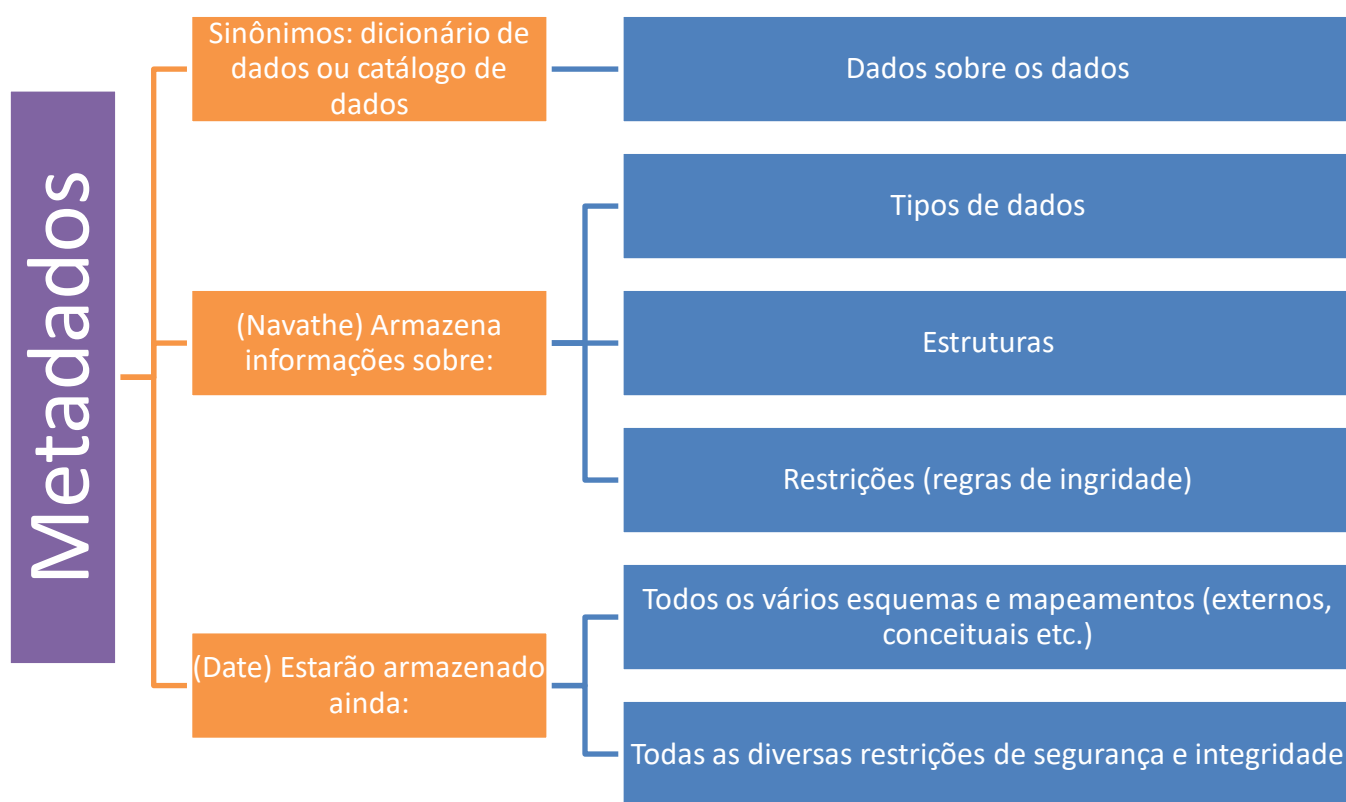
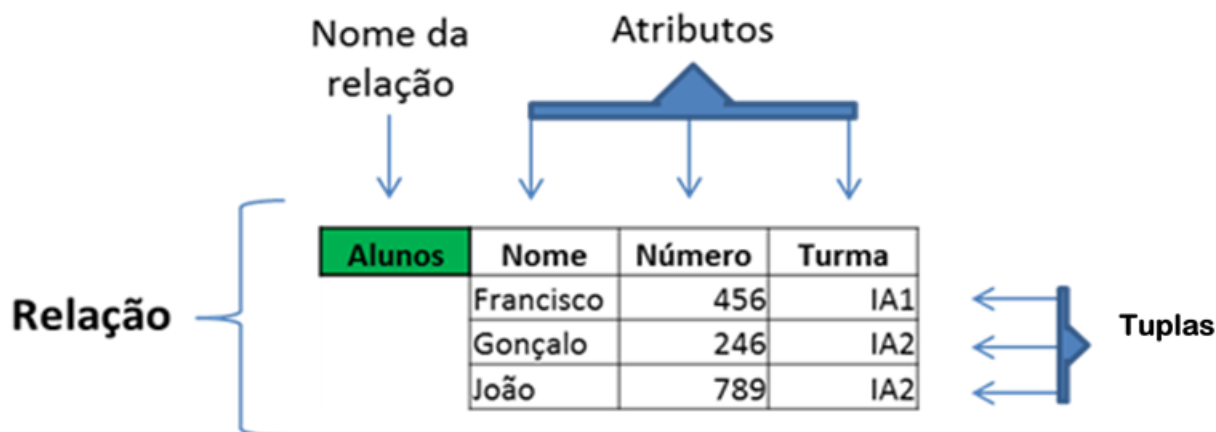


Figura 3 - Metadados, catálogo de dados ou dicionários de dados

TABELAS

Cada linha de uma tabela é conhecida como uma **tupla**, ou uma coleção de valores relacionados. Cada coluna é vista como **um atributo**, que possui um determinado conjunto de valores possíveis: o **domínio**. Um domínio é um conjunto de valores atômicos. A figura a seguir resume os conceitos vistos por meio da representação da **tabela Alunos**.



Percebam que a tabela representa uma relação dentro do modelo de dados. Cada uma das colunas (Nome, Número e Turma) representa um atributo, que deve ser associado a cada uma das tuplas desta tabela. Cada linha descreve um aluno desta tabela. A primeira linha apresenta os dados de Francisco com seus respectivos número e turma.

Outra constatação que podemos fazer, a partir da figura anterior, é a presença de um conjunto de valores possíveis para cada uma das colunas. Não está explícito. Não sabemos a definição formal da tabela. Contudo, podemos observar que as tuplas desta relação apresentam apenas valores numéricos associados a coluna Número e caracteres associados a coluna Nome. É importante perceber que cada coluna possui um **tipo de dados**.



Um **tipo de dados** define os valores que um dado pode assumir e as operações que podem ser efetuadas sobre ele. Tipos podem ser, por exemplo: inteiros, reais, caracteres etc.

Quando especificamos **um domínio**, geralmente definimos um tipo de dado do qual são retirados os valores possíveis para o atributo. Além do tipo de dado, um domínio pode ser descrito por meio de um nome, um formato e outras informações adicionais a respeito dos dados. Apenas para exemplificar, um atributo **salário** pode ser definido como **um decimal positivo com 10 dígitos e seu valor ser definido em reais (R\$)**.

Outra característica relevante associada ao domínio é o fato dele **ser atômico**. Isso **faz parte da definição do modelo relacional**. Não se esqueça disso! Outro ponto importante está relacionado à **integridade dos dados**. O modelo apresenta um conjunto de restrições básicas para dados e relacionamentos, conhecidas como **restrições de integridade**.



Ok! Temos as nossas relações! Conseguimos definir, pelo menos de forma gráfica, um conjunto de tabelas com seus atributos. Se quisermos, podemos povoar ou inserir tuplas nessas tabelas com seus respectivos dados. Agora, se eu quiser associar ou analisar os dados de mais de uma tabela conjuntamente, ou ainda, fazer uma consulta que me retorne apenas as tuplas ou linhas com uma certa restrição, o que devo fazer?

Os **operadores relacionais** são usados para manipular as informações pertencentes a cada uma das relações do nosso modelo. Esse conjunto de operações é conhecido como **álgebra relacional**. Ela vai nos ajudar a trabalhar com os dados em nossos esquemas relacionais.

Você deve estar lembrado que o esquema é considerado o projeto do banco de dados, a forma de bolo sobre a qual nossas instâncias são armazenadas. Assim, imagine uma tabela aluno com 100 alunos. O esquema vai definir os atributos necessários para cada linha da relação. Já as linhas são consideradas instâncias da tabela. Lembrando que as linhas podem ser denominadas **tuplas da tabela**. Que tal falarmos um pouco mais sobre elas? Antes ... vamos fazer uma questão ...



(Ministério da Economia – Especialista em Ciência de Dados - 2020) Julgue os itens a seguir, a respeito de banco de dados relacionais.

Um esquema de banco de dados é um conjunto de regras que governa um banco de dados ou todo o conjunto de objetos pertencentes a determinado usuário.

Comentários: O esquema é um outro nome para o projeto de banco de dados. Se pensarmos no modelo relacional, o esquema vai definir os nomes das colunas, seus tipos de dados e restrições de integridade. Por mais estranho que pareça, tanto as restrições quanto os tipos de dados são considerados regras que orientam a formação e os conteúdos válidos que irão aparecer em uma tupla.

Gabarito: CERTO.

Uma tupla deve conter um conjunto de elementos. Cada elemento deve conter o **nome** do atributo, o nome do **tipo** e um **valor**. Por exemplo:

TUPLA {**NOME:CARACTER:THIAGO**, **CARGO:CARACTER:PROFESSOR**}

Veja que cada um dos elementos do conjunto acima, por exemplo, **NOME:CARACTER:THIAGO**, corresponde a um **componente** da tupla e é conhecido como **tripla (conjunto de 3 valores) ordenada**. A quantidade de componentes de uma tupla é conhecida como **grau** ou aridez da tupla. Neste caso, o grau da nossa tupla é igual a 2.

Os dois primeiros valores, **NOME:CARACTER**, são vistos como um par ordenado e descrevem um atributo da tupla. Neste exemplo, **THIAGO** é o **valor** do atributo em questão. O conjunto completo de atributos é conhecido como **cabeçalho**, este vai determinar o **tipo de uma tupla**.



Segundo o CJ Date, as tuplas satisfazem a uma série de propriedades importantes, todas elas consequências imediatas das definições apresentadas até aqui:

- (a) Cada tupla **contém exatamente um valor para cada um de seus atributos**.
- (b) Não existe ordenação da esquerda para a direita nos componentes de uma tupla. Essa propriedade acontece porque uma tupla é definida por meio de um conjunto de componentes, e os conjuntos na matemática não possuem qualquer ordenação em seus elementos. Essa é a opinião do CJDate, como falamos acima.



Para tudo e presta atenção!!

Existe uma divergência na literatura quanto a ordenação dos componentes em uma tupla. O Codd, no seu artigo original que estabeleceu os conceitos do modelo relacional, afirma que **a ordem das colunas é significativa**.

O Elmasri segue nessa mesma linha, afirmando que uma tupla é uma lista ordenada de valores, de modo que a ordem dos valores em uma tupla — e, portanto, dos atributos em um esquema de relação — **é importante**.

- Entretanto, o mesmo Elmasri complementa seu texto com a seguinte frase: “... em um nível mais abstrato, a ordem dos atributos e seus valores não é tão importante, desde que a correspondência entre atributos e valores seja mantida.”¹

- (c) Cada subconjunto de uma tupla é uma tupla (e cada subconjunto de um cabeçalho é um cabeçalho). Mais que isso, essa afirmação também é válida para um subconjunto vazio em particular!

Esse último ponto merece uma explicação mais detalhada. Existe a tupla vazia (**TUPLA { }**). Às vezes, referimo-nos a uma tupla de grau zero mais explicitamente como uma “**0-tupla**”, a fim de enfatizar o fato de que ela não possui componentes. Outro nome conhecido é de tupla **nulária**.

Antes de continuar, vamos resolver mais uma questão:



¹ Elmasri, Ramez. Sistemas de Bancos de Dados (p. 42). Edição do Kindle.

Ano: 2019 Banca: TRC Órgão: Estratégia Assunto: Modelo Relacional - Tuplas

Analise as afirmações abaixo:

I. Cada tupla contém exatamente um valor (do tipo apropriado) para cada um de seus atributos.

II. Existe ordenação da esquerda para a direita nos componentes de uma tupla, basta observarmos uma tabela desenhada em um papel para constar tal fato.

III. Cada subconjunto de uma tupla é uma tupla (e cada subconjunto de um cabeçalho é um cabeçalho).

Quanto às propriedades das tuplas, é correto o que consta APENAS em

A I., II. e III.

B II.

C III.

D I e II.

E I e III.

Comentário: Acabamos de aprender sobre as propriedades de uma tupla. Perceba que as afirmações I e III são substratos das propriedades (a) e (c) descritas acima. Já a afirmação II mostra exatamente a minha preocupação com o seu entendimento de que uma relação não é exatamente uma tabela. A relação é um objeto formal, enquanto a tabela é um objeto informal que pode ser representada em uma folha de papel. Vejamos algumas das diferenças entre tabelas e relações:

Característica	Relação	Tabela
Nomes de tipo são omitidos nos cabeçalhos.	Não	Sim (normalmente)
Nomes de tipo e do atributo normalmente são omitidos no corpo.	Não	Sim
Atributos/colunas possuem uma ordem da esquerda para a direita. (Releia a explicação do quadro da página 10)	Não	Sim
As tuplas/linhas possuem uma ordem de cima para baixo.	Não	Sim
Pode conter tuplas/linhas duplicatas.	Não	Sim ²

Assim, podemos avaliar a segunda afirmação como falsa e achar nossa resposta na alternativa E.

Gabarito: E

Falamos das propriedades importantes das tuplas. Neste momento, vamos falar das propriedades importantes das relações. Dentro de qualquer relação, temos as seguintes propriedades de acordo com CJDate:

² Neste caso, pense em uma tabela do Excel. Se estivermos falando de uma tabela no modelo relacional, ela não pode conter linhas duplicadas e todos os seus atributos devem ser atômicos.



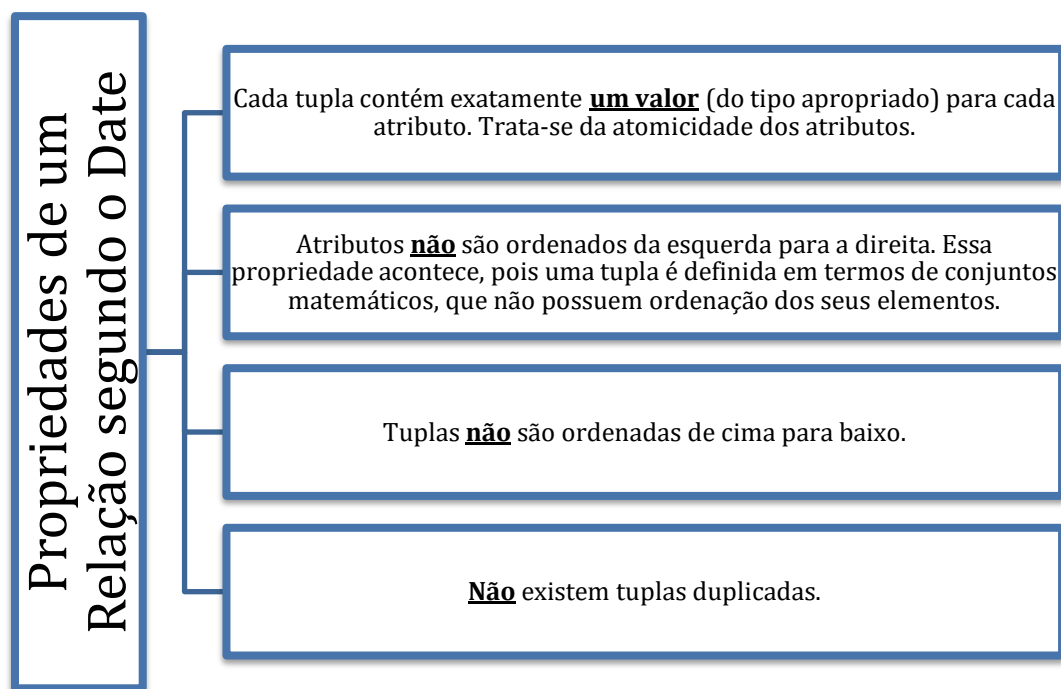


Figura 4 - Propriedades de uma relação.

Buscando praticar os conceitos adquiridos até aqui, vamos fazer uma questão do CESPE sobre o assunto e mostrar que estamos alinhados com o conteúdo que vem sendo cobrado em concursos públicos.



Ano: 2018 Banca: CESPE Órgão: TCM-BA Cargo: Auditor de Contas
Questão: 09

Considerando os conceitos de banco de dados relacionais, assinale a opção correta a respeito das propriedades de uma tupla.

- A A tupla tem o mesmo significado e as mesmas propriedades de uma tabela.
- B Os componentes de uma tupla são ordenados da esquerda para a direita.
- C Cada tupla contém exatamente um valor para cada um de seus atributos.
- D Um subconjunto de uma tupla não é considerado uma tupla.
- E Uma tupla nunca é vazia, seu grau pode variar de 1 até n.

Comentário: Vimos que as tuplas são linhas de uma tabela. Elas, quando consideramos a referência teórica e matemática do modelo relacional, não são ordenadas e não se repetem. Vejam que a questão acima usou como referencial teórico o livro do Date. Vamos agora analisar cada uma das alternativas acima:

A) A tupla é a linha da tabela. Representa uma instância ou um valor armazenado. Pense na tabela aluno, cada aluno armazenado é uma tupla da tabela. Contudo, uma tabela pode ter atributos que não são específicos da tupla, como a quantidade máxima de registros. Logo, não podemos definir o todo pela parte, nem podemos dizer que a forma do bolo é um bolo. Sendo assim, a alternativa está **incorreta**.

B) As tuplas **não** são ordenadas de cima para baixo, **nem** seus atributos ou componentes são ordenados da esquerda para a direita.



C) Pela definição de modelo relacional do Codd, os atributos de uma relação devem ter valores atômicos. Logo, cada coluna deve ter um valor dentro do seu respectivo domínio. Este pode ou não aceitar valores nulos. Sendo assim, essa afirmação está certa.

D) Um subconjunto de uma tupla é outra tupla. Inclusive, se lembrarmos da propriedade de fechamento das operações de álgebra relacional, podemos perceber que o resultado de uma projeção vai reduzir as tuplas de uma tabela aos atributos definidos na operação.

E) Uma tupla de uma tabela pode ser vazia, ela é conhecida como *empty tuple* ou *nullary tuple*. Para preencher valores desconhecidos, usamos o valor nulo, que é uma notação para vazio. Além disso, ela representa um conjunto de atributos. A quantidade de atributos de uma tupla é denominada grau e pode variar de 0 até n. Logo, temos mais uma alternativa incorreta.

Gabarito: C

VISÕES

As views ou visões são objetos SQL que podemos criar dentro dos nossos bancos de dados. Uma VIEW é um **comando SQL que é armazenado no banco de dados** e possui um nome associada a ela. Podemos observar algumas funções básicas. A primeira é facilitar a visualização dos dados dispersos em diversas tabelas, tornando-os mais natural ou intuitivo ao entendimento humano.

Outra função importante para a *view* está relacionada à segurança dos dados. É possível restringir o acesso aos campos e às colunas de uma tabela por meio de uma *view*. Desta forma, o usuário teria visão apenas a parte dos dados ou das informações. Esse grupo de informações deve ser compatível com as funções e as necessidades de acesso do usuário.

Uma terceira opção para o uso de *view* é sumarizar dados de diferentes tabelas, gerando relatórios. Vejamos abaixo dois exemplos do uso de **Views**. Lembrando que ela pode ser criada sobre uma ou múltiplas tabelas. Observe que o comando, basicamente, inclui a sintaxe CREATE VIEW nome AS antes de uma consulta ao banco de dados.

```
CREATE VIEW profs_estrategia AS  
SELECT pf.primeironome, pf.ultimonome, pf.telefone, pf.email  
FROM professores pf  
NATURAL JOIN disciplina d  
WHERE d.nome = 'Informática';
```

Para visualizarmos os dados de uma visão, basta escrevermos um comando SELECT sobre ela, vejam o exemplo sobre a view **profs_estrategia** criada acima.

```
SELECT * FROM profs_estrategia;
```

A view é considerada uma tabela virtual, porque ela só existe durante o período que você está utilizando-a. Todas as operações que são feitas sobre a tabela podem ser feitas em uma VIEW, mas a tabela é virtual e, na teoria, não deve ser armazenada no banco de dados.



Vamos comparar os conceitos de tabelas e visões de forma pragmática. A distinção entre uma **tabela básica** e uma visão é constantemente caracterizada desta maneira:

Tabelas básicas “existem realmente”, no sentido de que representam dados de fato armazenados no banco de dados.

As visões, ao contrário, não “existem realmente”, mas apenas oferecem diferentes modos de visualização dos “dados reais”.



(Ministério da Economia – Especialista em Ciência de Dados - 2020) Julgue os itens a seguir, a respeito de banco de dados relacionais.

Uma view é uma tabela que é atualizada no momento em que uma das tabelas consultadas é atualizada; a view permite consultas ao banco de dados de forma mais rápida quando comparada à utilização de índices.

Comentários: Uma visão é uma tabela temporária. Ela é carregada no momento em que o usuário acessa a visão. Dependendo da sua estrutura (da consulta que vou utilizada para extrair os dados) essa visão pode sofrer atualização e essas atualizações podem ser refletidas nas tabelas subjacentes. As atualizações nas tabelas não são refletidas na visão imediatamente. A visão não necessariamente melhora a performance do banco de dados e não faz sentido compará-la a um índice.

Gabarito: ERRADO.

OK! Até aqui falamos que a visão é uma tabela virtual que não é armazenada fisicamente. Contudo, alguns SGBDs utilizam o conceito de visão materializada (também conhecida como snapshot), ou seja, eles guardam os registros de uma visão em um arquivo no disco. Isso traz um benefício de performance. Para efeitos de concursos públicos, você só deve considerar a existência deste tipo de visão se a banca deixar explícito no enunciado.

Outro ponto interessante é o seguinte: a visão é uma consulta feita sobre dados de banco de dados e carregados na memória! Isso você já sabe ... mas, depois de carregar os dados na memória, você pode manipulá-lo, inclusive fazendo atualização nos dados da visão. Daí eu pergunto, essas atualizações serão refletidas nas tabelas subjacentes (as tabelas que são usadas como referência para a consulta)?

A resposta é: DEPENDE!! Como assim? Se cada linha da visão estiver relacionada a uma linha da tabela subjacente, as atualizações serão refletidas na base de dados, da mesma forma que as atualizações na tabela subjacente devem ser repassadas a visão. Contudo, caso exista alguma função de agregação ou agrupamento (GROUP BY ou HAVING), ou a visão seja definida sobre várias tabelas com consultas aninhadas, a complexidade impede a atualização das tabelas base.

Isso nos leva a uma classificação bastante difundida das visões: elas podem ser simples ou complexas. A visualização simples é criada a partir de apenas uma única tabela e não possui nenhuma função. As operações de manipulação podem ser executadas sobre



essas visões. Já as visões **complexas** contêm mais de uma tabela base ou é criada a partir de mais de uma tabela, podem possuir funções agregadas e grupos de dados. Não é possível aplicar operações de manipulação diretamente sobre visões complexas.

Agora vamos colocar essas definições em um esquema:

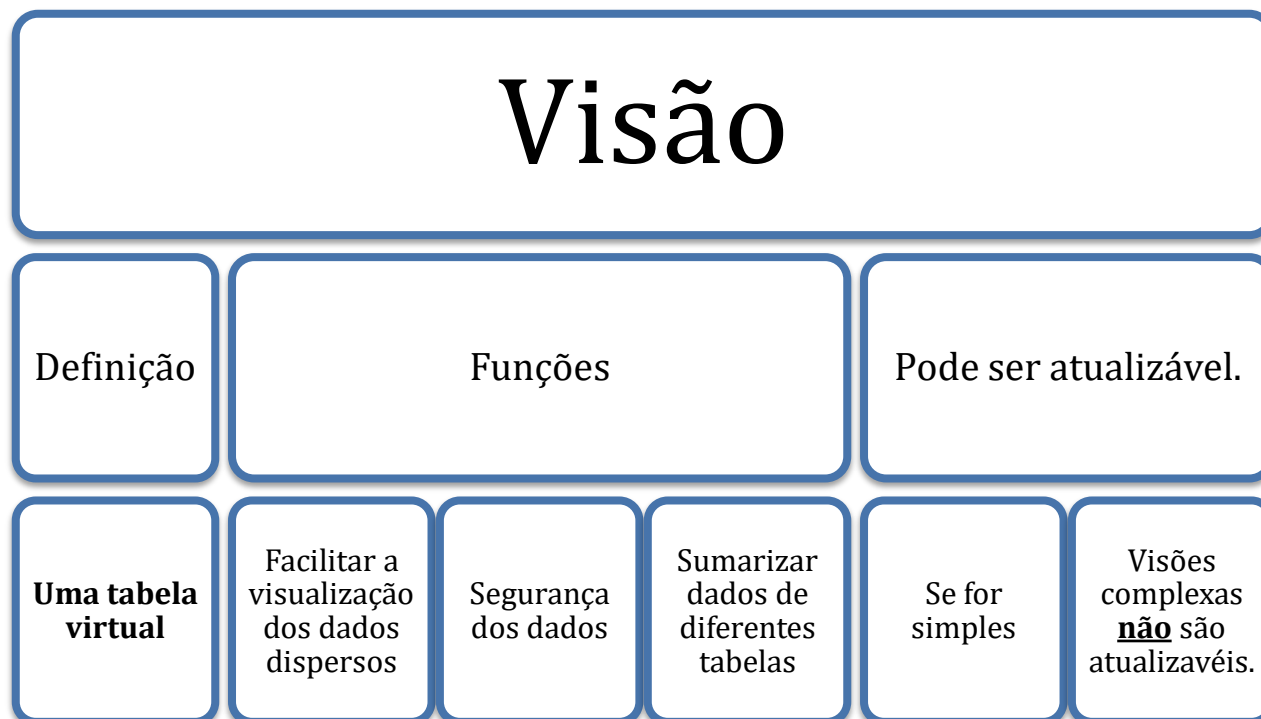


Figura 5 - Conceitos de Visão.

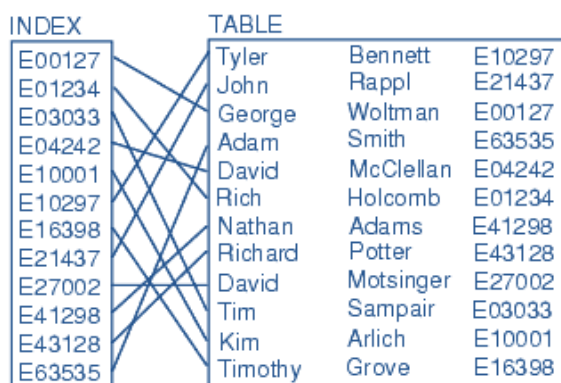
ÍNDICES

O primeiro conceito que devemos ter em mente sobre o assunto é a definição de índices. **Um índice é um mecanismo utilizado para melhorar a velocidade de acesso aos dados.** Ele é composto por uma chave, que é um atributo ou um conjunto de atributos usado para procurar registros em um arquivo. Um índice também possui um ponteiro, que consiste em um identificador para um bloco de disco, além do deslocamento dentro do bloco para encontrar o registro.

Um arquivo de índice consiste em um conjunto de registros com o formato apresentado na figura abaixo. A esse registro se dá o nome de registro de índice ou entrada de índice.



Como uma técnica para criar estruturas de dados auxiliares, os índices agilizam a busca e a recuperação de registros. Para isso, eles envolvem armazenamento de dados auxiliares. Esses dados são armazenados nos arquivos de índices. Alguns tipos de acesso podem se beneficiar dos índices, por exemplo, a localização de registro com um valor especificado e a localização de registros em um intervalo especificado de valores. Veja a figura abaixo para entender um pouco mais sobre índices:



É necessário saber se esse benefício, de fato, traz um ganho de desempenho ao sistema de banco de dados. Avaliar os índices para medir seus efeitos na performance é um passo importante. Vários SGBDs possuem utilitários que ajudam a quantificar os efeitos pretendidos com a criação de índices sobre tabelas. Esse valor é baseado em alguns fatores, entre eles, o tempo de acesso, inserção e deleção; o overhead de espaço em disco e os métodos de acesso suportados.

Um índice é dito **denso** quando existe uma entrada de índice no arquivo para cada valor de chave de pesquisa e, portanto, para cada registro no arquivo. Definimos um índice como **esparso**, quando temos entradas de índices apenas para alguns valores de pesquisa. Os índices primários e de *clustering* são categorizados como esparsos.

Os índices podem ainda ser classificados de acordo com a figura abaixo:



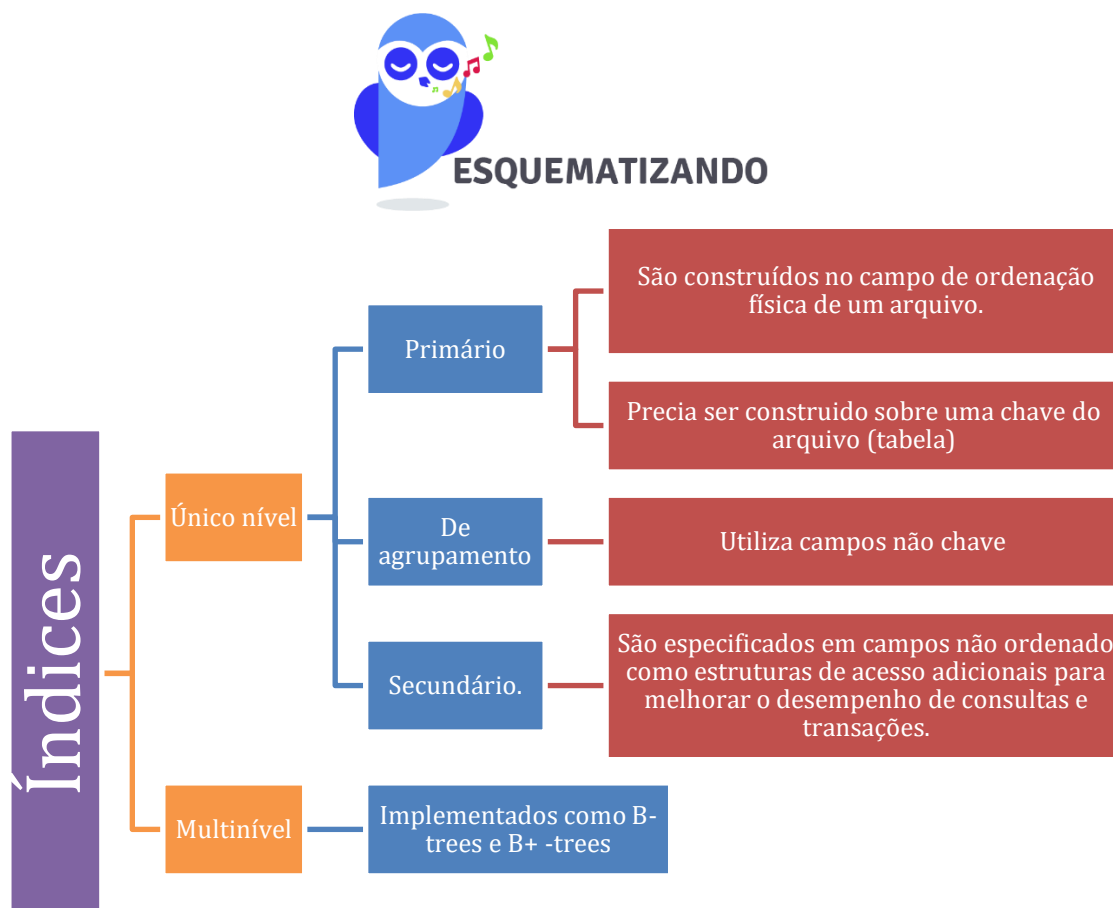
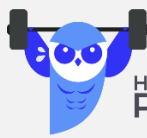


Figura 6 - Conceitos de índices



HORA DE
PRATICAR!

(Ministério da Economia – Especialista em Ciência de Dados - 2020) Julgue os itens a seguir, a respeito de banco de dados relacionais.

Os índices secundários precisam ser densos, com uma entrada de índice para cada valor de chave de busca e um ponteiro para cada registro no arquivo.

Comentários: Os índices secundários são criados sobre colunas que não seguem a mesma ordenação do arquivo. Desta forma, para cada entrada do arquivo de dados é necessário um registro no arquivo de índice, o que faz com que o índice seja considerado DENSO.

Gabarito: CERTO.



CHAVES E RELACIONAMENTOS

Superchave, chave, chave candidata, chave primária e chave estrangeira

Vamos começar falando sobre **chave**. Existem alguns conceitos relacionados à chave. O primeiro deles é o de **superchave**. Trata-se de uma coluna ou uma combinação de colunas contendo valores únicos para cada linha. A combinação de todas as colunas em uma tabela sempre é uma superchave, porque as linhas de uma relação devem ser sempre únicas. Em outras palavras, uma superchave é um conjunto de atributos que tem a característica de restringir o conjunto de tuplas de uma relação em apenas uma linha.

Quando olhamos para um conjunto de atributos em uma tabela que não se repetem em nenhuma das linhas da tabela, podemos considerá-los uma **superchave**.

Já entendemos o conceito de superchave. Uma superchave pode ser considerada apenas uma chave. Neste caso, a chave é defendida como uma **superchave mínima ou irreduzível** (vou chamá-la de K), onde qualquer remoção de atributo de K fará com que K deixe de ser superchave da relação. Se um esquema tiver mais de uma chave, cada uma delas é chamada de **chave candidata**. Entre as chaves candidatas, uma delas é escolhida para ser a chave da relação e é denominada **chave primária**. As demais são renegadas e são denominadas **chaves secundárias ou alternativas**.

Segundo o Date, se K for um conjunto de atributos de uma variável de relação R, então K é uma **chave candidata** para R, se e somente se, ela possui ambas as propriedades.



Unicidade: Nenhum valor válido de R contém duas tuplas diferentes com o mesmo valor para K.

Irreduzibilidade: Nenhum subconjunto apropriado de K tem a propriedade de unicidade.

De posse deste conhecimento, é possível definir o conceito de **atributo primário**, que nada mais é do que um atributo membro (que faz parte) de alguma **chave candidata** da relação (R). Por sua vez, de forma bem intuitiva, um **atributo não primário** é todo aquele que não for um atributo primário!

Ano: 2019 Banca: CESPE Órgão: SEFAZ-RS Prova: Auditor Assunto: Banco de Dados

No modelo relacional, a afirmação “Duas tuplas distintas, em qualquer estado da relação, não podem ter valores idênticos para os atributos na chave” é

A uma propriedade de chave do modelo.

B falsa.

C uma restrição de domínio do modelo.

D uma propriedade exclusiva do modelo objeto-relacional.



E uma condição que deverá estar explícita na representação dos atributos de uma tupla.

Comentário: As principais propriedades das chaves são unicidade e irredutibilidade. Vejamos as definições:

Unicidade: Nenhum valor válido de relação contém duas tuplas diferentes com o mesmo valor para os atributos chave.

Irredutibilidade: Nenhum subconjunto apropriado da chave tem a propriedade de unicidade.

Veja que a questão trata de unicidade, logo uma propriedade de chave do modelo.

Gabarito: A



Nesta parte da aula, gostaria que você fixasse o conceito de chave primária e estrangeira. Esse é um conceito importante dentro do assunto de banco de dados. A chave primária vai ser usada pelo SGBD na organização dos arquivos de dados. Isso significa que o acesso aos dados pode ser feito de forma mais rápida, usando o conjunto de atributos definido como chave primária.

O conjunto de atributos deve manter a propriedade de unicidade, ou seja, cada valor associado a uma chave primária só pode aparecer em uma linha da tabela. Olhe para a figura abaixo e me responda: quais campos podem ser usados como chave primária da tabela APROVADOS?

APROVADOS

<u>Id Aprovado</u>	Nome	Município	CPF	Passaporte
1	Thiago	2611606	11111111111	111111
2	Flávia	2604106	22222222222	222222
3	Vinícius	2604106	33333333333	333333
4	Lucas	Null	44444444444	444444



Atributos de uma relação cujos valores devem corresponder a valores de alguma chave candidata de alguma relação.

Chave candidata

Chave primária

Chave Alternativa

Chave Estrangeira

Toda superchave com a propriedade de irredutibilidade.

A chave candidata designada para identificar as linhas de uma tabela.

<u>Id Município</u>	Nome	Descrição
2611606	Recife	Veneza brasileira
2604106	Caruaru	Capital do forró

Figura 7 - Chaves e relacionamentos



Se você pensou nos atributos Id_Aprovado, CPF e Passaporte, parabéns! Você pensou certo. Qualquer um destes campos pode ser usado como chave primária da tabela, por isso, eles são denominados **chaves candidatas**. No nosso exemplo, escolhemos o atributo Id_aprovado para ser usado como **chave primária** da relação Aprovados e o campo Id_Município para ser chave primária da relação MUNICIPIO. As outras chaves que foram preteridas podem ser denominadas **chave alternativa**.

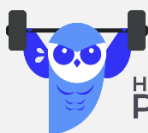
Agora, vamos falar sobre **chave estrangeira**. Observe que, na relação aprovados, temos um campo denominado Município. Esse campo vai “apontar” para a chave primária da tabela Município (Id_Município). O que acontece aqui é que os valores presentes na coluna Município da tabela aprovados devem estar presentes na chave da tabela município. Outra opção é que o valor seja definido como desconhecido (nulo). Qualquer outro valor que apareça na coluna Município fere a integridade referencial. Neste caso, é como se você estivesse se referindo a um município não cadastrado. Isso fere a integridade referencial que vimos anteriormente.

A chave estrangeira é considerada uma coluna ou uma combinação de colunas em que os valores devem corresponder aos valores de uma chave candidata. A chave estrangeira deve ter tipo de dado igual ao da chave candidata associada. As chaves estrangeiras são usadas no modelo relacional para construir relacionamentos entre as tabelas.

É possível, ainda, um relacionamento em que a chave estrangeira aponta para a mesma tabela. Neste caso, temos um auto relacionamento. Os auto-relacionamentos representam associações entre membros do mesmo conjunto. Os auto-relacionamentos não são comuns, mas são importantes quando ocorrem. No banco de dados de uma universidade, um professor pode supervisionar outros professores e ser supervisionado por um professor. Auto relacionamento também pode ser chamado de relacionamento recursivo ou unário (apenas uma entidade participa do relacionamento). Veja o exemplo de um auto relacionamento na tabela abaixo:

Auto-relacionamento

CPFProf	NomeProf	SobrenomeProf	CidadeProf	UFProf	DeptoProf	ClassificacaoProf	SalarioProf	SupervisorProf	DataAdmProf	CEPProf
098-76-5432	LEONARD	VINCE	SEATTLE	WA	ADM	ASSISTENTE	R\$35.000	654-32-1098	01-Abril-95	98111-9921
543-21-0987	VICTORIA	EMMANUEL	BOTHELL	WA	ADM	CATEDRATICO	R\$120.000		01-Abril-96	98011-2242
654-32-1098	LEONARD	FIBON	SEATTLE	WA	ADM	ASSOCIADO	R\$70.000	543-21-0987	01-Abril-95	98121-0094
765-43-2109	NICKI	MACON	BELLEVUE	WA	FINAN	CATEDRATICO	R\$65.000		01-Abril-97	98015-9945
876-54-3210	CRISTOPHER	COLAN	SEATTLE	WA	ADM	ASSISTENTE	R\$40.000	654-32-1098	01-Abril-99	98114-1332
987-65-4321	JULIA	MILLS	SEATTLE	WA	FINAN	ASSOCIADO	R\$75.000	765-43-2109	01-Abril-00	98114-9954



HORA DE
PRATICAR!

(Ministério da Economia – Especialista em Ciência de Dados - 2020) Julgue os itens a seguir, a respeito de banco de dados relacionais.

Um banco de dados relacional organiza os dados em tabelas e os vincula, com base em campos-chave, e essas relações permitem recuperar e combinar dados de uma ou mais tabelas com uma única consulta

Comentários: Exatamente! O princípio da informação diz que tudo no banco de dados relacional é representado por atributos em relações. Os relacionamentos seguem essa mesma lógica. Para representar os relacionamentos usamos chaves estrangeiras. Os atributos também são usados como parâmetros para recuperação das informações e para executar a junção entre duas tabelas. Logo, temos uma alternativa correta.

Gabarito: CERTO.



(Ministério da Economia – Especialista em Ciência de Dados - 2020) Julgue os itens a seguir, a respeito de banco de dados relacionais.

Em um banco de dados relacional, a chave candidata a primária é formada por um ou mais atributos que identificam a única tupla.

Comentários: A ideia é que todo conjunto de atributo que tenha as propriedades de unicidade e irredutibilidade seja considerado chave. Numa relação com mais de uma chave, temos um conjunto de chaves candidatas a chave primária. Em cada relação, apenas um conjunto de atributos pode ser escolhido como chave primária. Toda chave identifica unicamente cada uma das linhas ou tuplas de uma relação.

Gabarito: CERTO.



(Ministério da Economia – Especialista em Ciência de Dados - 2020) Julgue os itens a seguir, a respeito de banco de dados relacionais.

A restrição de integridade referencial exige que os valores que aparecem nos atributos especificados de qualquer tupla na relação referenciadora também apareçam nos atributos de pelo menos uma tupla na relação referenciada.

Comentários: Essa questão encheu meus olhos de lágrimas. A chave estrangeira é um atributo (ou conjunto de atributos) cujo valor é limitado por um outro atributo ou coluna da relação referenciada. Da forma como está escrito, conseguimos inclusive visualizar os casos em que a chave estrangeira aponta para a própria tabela, o caso do relacionamento gerencia, onde os valores da coluna CPF do gerente é um valor válido na coluna CPF da mesma relação. Parabéns para o CESE!! Usar os termos relação referenciada e referenciadora foi sensacional!!

Gabarito: CERTO.



RESTRIÇÕES DE INTEGRIDADE

Nas partes anteriores desta aula, definimos a estrutura tabular presente no modelo relacional. Agora vamos falar das restrições de integridade. Essas restrições visam **manter a consistência dos dados dentro do seu banco**. As regras de integridade podem ser vistas como um conjunto de parâmetros ou regras do negócio, previamente estabelecidos e criados no banco de dados, aos quais os dados são submetidos. As regras servem para garantir que um processo de atualização não resulte em dados inconsistentes.

As **restrições de integridade** resguardam o Banco de Dados **contra danos acidentais**, assegurando que mudanças feitas **por usuários autorizados** não resultem na perda de consistência de dados.

Uma das características mais fortes dos bancos de dados relacionais, quando comparados com seus antecessores (em Rede e Hierárquico), está em oferecer mecanismos para a criação de regras de integridade **diretamente no banco de dados**. As regras de integridade de dados podem ser implementadas nos SGBDs de forma **declarativa** ou **procedural**.

A **integridade declarativa** é implementada através de parâmetros opcionais da linguagem de definição de dados (DDL). Os tipos mais comuns de integridade declarativa são: chave primária (PK), domínio e integridade referencial.

A **integridade procedural** se apresenta sob a forma de um programa, cuja lógica é escrita pelo programador, na linguagem procedural nativa do SGBD. Esse tipo de integridade supre as necessidades não cobertas pelos parâmetros de integridade declarativa e pode ser criada através de *triggers* (gatilhos), *stored procedures* (procedimentos armazenados) ou *assertions* (afirmações).

Triggers, *stored procedures* e *assertions* são comandos que podem ser usados para fazer ajustes na base de dados. Um trigger, por exemplo, é uma estrutura conhecida como EVENTO-CONDIÇÃO-AÇÃO. Imagine que uma modificação no banco de dados dispara um evento. Por exemplo, uma atualização na tabela funcionário com a inclusão de 10 novos servidores.

Esse evento vai levar à execução de um código ou de um programa. O programa vai verificar (condição) se os servidores foram de fato aprovados no concurso público e, em seguida, incluir os dados deles na folha de pagamento e no plano de saúde do órgão em questão. Eis que descrevemos de forma abstrata a execução de um TRIGGER. As bancas focam suas questões sobre esse assunto nos seguintes tipos de restrições de integridade:

Integridade de Domínio (dom(A)) – restringe os valores válidos que podem ser associados a um determinado atributo. **É a mais elementar forma de restrição de integridade**. Ajudam, não somente a garantir os valores inseridos no banco de dados, mas também a testar consultas para garantir que as comparações feitas fazem sentido.



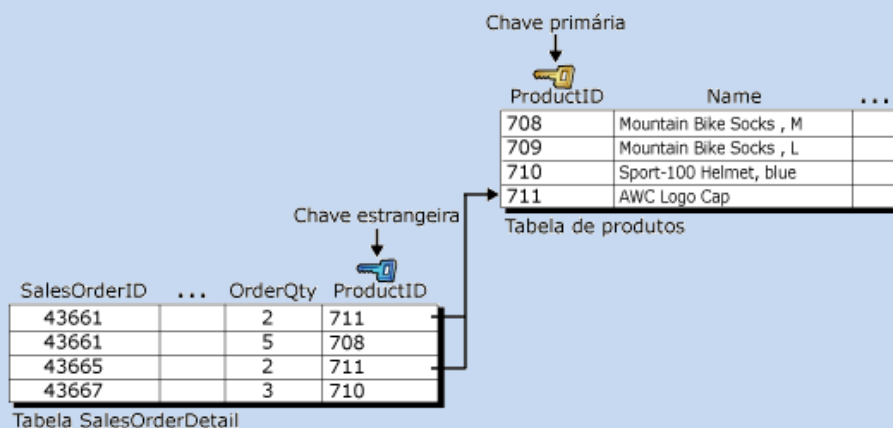
Você deve se lembrar que cada tipo de dado tem um conjunto de operações que pode ser executado sobre eles. Imagine as operações de soma e subtração nos tipos de dados numéricos. Na linguagem SQL, é implementada pelo **comando CHECK**.

Integridade de Chave (Unicidade) – Garante a unicidade do valor da chave primária em cada uma das tuplas de uma relação. Implementado pela palavra-chave **UNIQUE**.

Integridade de Vazio (def if(x!=null?x==null)) – Basicamente verifica se um valor de um determinado atributo pode ou não ter o valor nulo associado a suas instâncias. Podemos definir uma coluna como **NULL** ou **NOT NULL** em SQL.

Integridade de Entidade (PK != null) – Afirma que cada tabela deve ter uma chave primária e garante que a chave primária de uma entidade não receba o valor nulo.

Integridade Referencial (FK == PK (Chave Candidata) || FK == null) - Garante que o valor que aparece em uma relação para um dado conjunto de atributos também apareça para um conjunto de atributos em outra relação. Em SQL, é implementada por meio de uma referência da chave estrangeira (FK) de uma relação à chave primária ou atributo UNIQUE da outra tabela. Esse valor de chave estrangeira, sempre que existir, deve estar associado a um valor da chave primária da outra relação. Caso contrário, pode assumir apenas o valor nulo. Veja a figura abaixo para entender melhor esse conceito:



Observe que o uso de chaves estrangeiras vai permitir a criação de relacionamentos no modelo relacional.

Integridade Semântica (*assertions* ou *triggers*) – Uma asserção é um predicado que expressa uma condição desejada a ser sempre satisfeita no banco de dados. Um gatilho (*triggers*) é um comando que é executado pelo sistema automaticamente, em consequência de uma modificação no banco de dados. Toda vez que um evento acontece, uma condição (caso exista) é verificada e uma ação é disparada.

Ao lado da explicação de cada uma das regras de integridade, existe, entre parênteses, uma palavra, uma fórmula ou uma expressão. Elas não existem em nenhuma fonte oficial



de estudos. São estruturas criadas por mim para facilitar a memorização das restrições de integridade. Quando estudamos a linguagem SQL, aprofundamos um pouco mais sobre esse assunto com uma visão mais prática dos conceitos apresentados.

Antes de continuarmos a tratar do assunto, vamos fazer uma rápida questão. Essa é uma questão inédita, criada pela banca TRC para trabalhar pontos específicos do assunto.

Ano: 2019 Banca: TRC Órgão: Estratégia Concursos Assunto: Restrições de integridade

Sobre as restrições de integridade presentes no modelo de dados relacional, analise as afirmações abaixo. Em seguida, assinale a alternativa que aponta a(s) correta(s).

I. A restrição de integridade de entidade estabelece que nenhum valor de chave primária pode ser null. Isso porque o valor da chave primária é usado para identificar as tuplas individuais em uma relação.

II. Todas as restrições de integridade deveriam ser especificadas no esquema do banco de dados relacional, caso queiramos impor essas restrições aos estados do banco de dados.

III. Ter valores null para chave primária implica não podermos identificar alguma tupla.

IV. A restrição de integridade referencial é classificada entre duas relações e é usada para manter a consistência entre as tuplas nas duas relações.

a) I, II, III e IV. b) I, II e III, apenas c) I, III e IV, apenas d) II, III e IV, apenas e) Apenas I

Comentário: A questão trata das restrições de integridade, que são utilizadas para garantir a consistência dos dados e a inviolabilidade em um banco de dados relacional. Podemos citar, como exemplos de restrição de integridade, a integridade de entidade, a integridade de vazio, a integridade de chave, a integridade referencial e a integridade semântica. Algumas dessas restrições são citadas nas alternativas. Vamos avaliar cada uma delas:

Alternativa I - Restrição de entidade - define que **nenhuma chave primária deve aceitar o valor nulo**. Caso contrário, poderíamos ter várias tuplas com chave primária contendo o mesmo valor, ou seja, o valor nulo. Isto impediria que pudéssemos identificar essas tuplas de forma única. **Alternativa correta.**

Alternativa II - As **restrições de integridade** devem ser implementadas no próprio banco de dados. Dessa forma, o SGDB pode, utilizando seus mecanismos internos, garantir, com eficiência, a consistência dos dados. Outros tipos de restrições, relacionadas às regras de negócio, serão implementadas na aplicação. **Alternativa correta.**

Alternativa III - Lembre-se que nulo é um valor que não consegue identificar nada. Ele é usado para descrever que o valor da variável em questão é desconhecido ou inexistente. O atributo número do apartamento da relação endereço pode ser nulo quando se referir ao endereço de uma casa. **Alternativa correta.**

Alternativa IV - A integridade referencial é o tipo de integridade que provê a consistência de dados relacionados entre duas ou mais tabelas. Especificamente, refere-se à chave estrangeira em uma tabela e à chave primária correspondente em outra tabela. Também se refere a quais ações o SGDB deverá executar nos dados associados às chaves estrangeiras quando ocorrerem mudanças (exclusão, alteração) na chave primária relacionada. **Alternativa correta.**

Gabarito: A

Para finalizar nosso conteúdo de restrições de integridade, brindaremos ao som de mais um mapa mental!



MAPA MENTAL – RESTRIÇÕES DE INTEGRIDADE

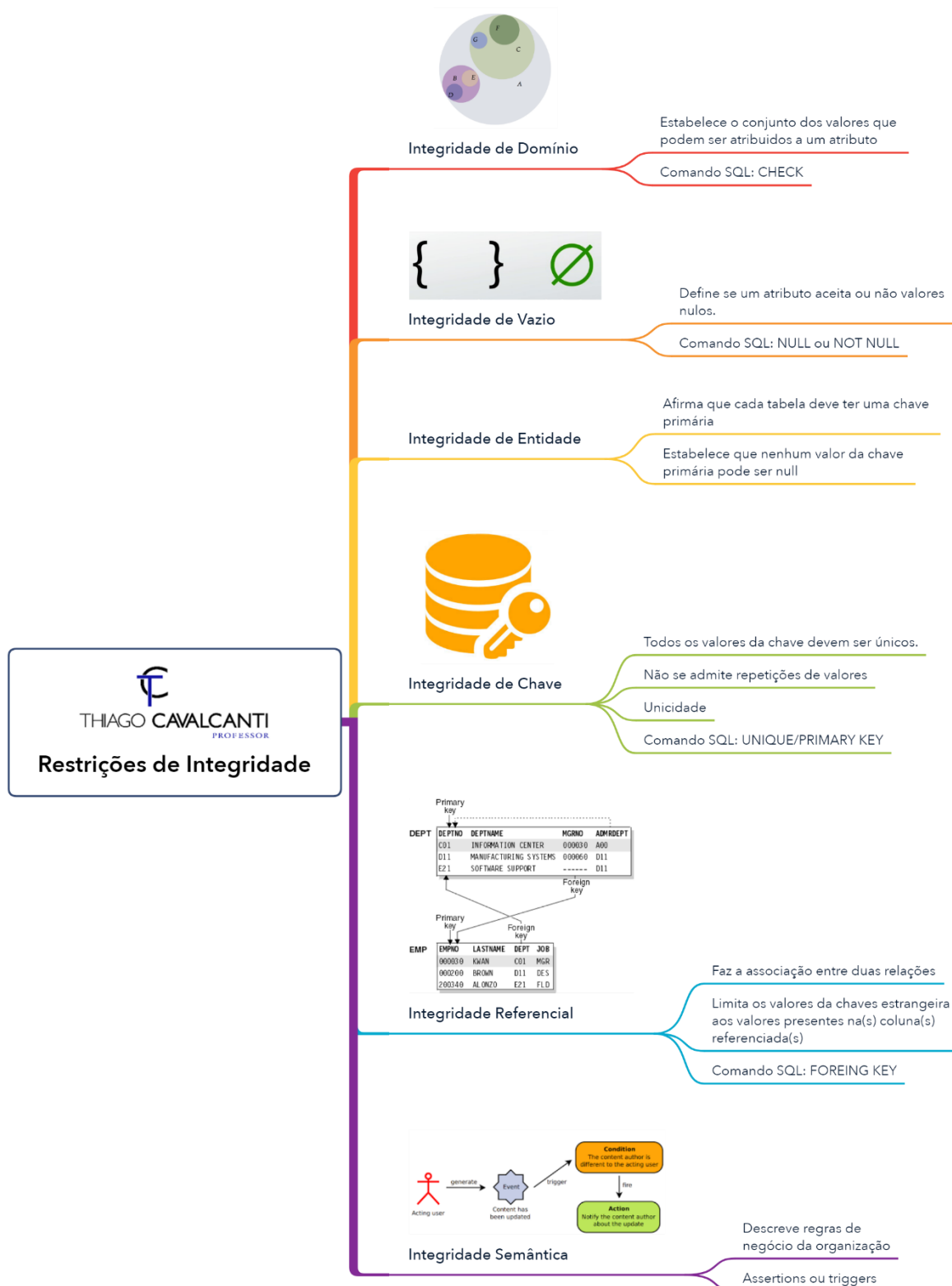


Figura 8 - Mapa mental de restrições de integridade



REGRAS DE CODD

Para que um banco de dados seja considerado relacional, ele deve seguir as **13 regras definidas** por E.F.Codd. Apresentamos abaixo as referências a cada uma das regras. É muito comum encontrar publicações falando das 12 regras. O que acontece é que são 13 regras, numeradas de 0 até 12.

0. **Regra fundamental ou regra base:** Todas as regras se baseiam na noção de que, para um banco de dados ser qualificado como relacional, ele deve utilizar recursos exclusivamente relacionais para seu gerenciamento.

1. **Informação:** Todas as informações em um banco de dados são **representadas** de forma explícita **no nível lógico** e são estruturadas exatamente, em apenas uma forma, por **valores em tabelas**. Em outras palavras, todos os valores em bases de dados relacionais são representados em colunas e linhas de uma tabela.

2. **Acesso Garantido:** Todos os dados precisam ser acessíveis. Cada valor atômico (**datum**), em um banco de dados relacional, possui a garantia de ser logicamente acessado pela combinação do nome de tabela, do valor da chave primária e do nome da coluna. Ou seja, os dados devem possuir identificadores únicos que permitam o acesso aos mesmos, evitando problemas de interpretação da informação.



3. **Tratamento sistemático de valores nulos:** Valores nulos ou em branco devem ser suportados de forma sistemática, independentemente do tipo de dado, para representar informações inexistentes e informações inaplicáveis. É também implícito que tais representações devem ser manipuladas pelo SGBD de maneira sistemática.

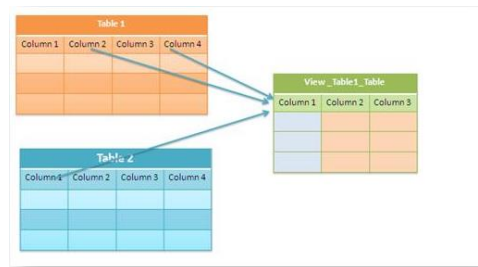
4. **Catálogo on-line dinâmico:** A **descrição ou metadados** do banco de dados é representada no nível lógico da mesma forma que os dados ordinários, permitindo que os usuários autorizados utilizem a mesma linguagem relacional aplicada aos dados regulares para terem acesso aos mesmos. É possível usar a linguagem SQL para conhecer as descrições das tabelas do banco, por exemplo. Veja a descrição de um conjunto de tabelas de um banco de dados na figura abaixo:

5. **Sublinguagem ampla dos dados:** Um sistema relacional pode suportar várias linguagens e várias formas de recuperação de informações. Entretanto, deve haver pelo menos uma linguagem, com uma sintaxe bem definida e expressa por um conjunto de caracteres, que suporte de forma compreensiva todos os seguintes itens:

- Sintaxe linear
- Acesso de forma interativa e/ou por meio de programas
- Operações de definição de dados, definição de "views", manipulação de dados (interativa e embutida em programas), restrições de integridade, autorizações/segurança de acesso.
- Administração de transações (begin, commit e rollback).



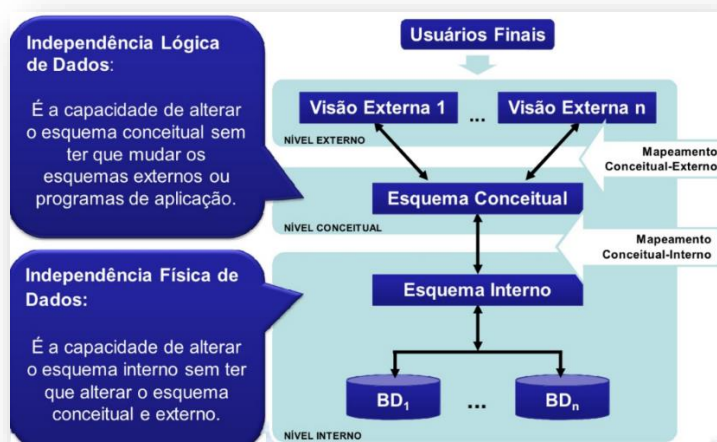
6. **Atualização de visualizações:** Todas as "views" ou visões dos dados que são teoricamente atualizáveis devem também ser atualizáveis pelo sistema. Lembrem-se que as visões nem sempre podem ser atualizáveis, ou seja, uma alteração na visão nem sempre gera uma mudança de estado nas tabelas associadas.



7. **Inserção, atualização e exclusão de alto nível:** A capacidade de manipular um conjunto de dados (em uma relação) através de um simples comando deve se estender às operações de inclusão, alteração ou exclusão de dados.

8. **Independência física de dados:** Programas de aplicação e recursos *ad hoc* permanecem logicamente inalterados quando ocorrem mudanças no método de acesso ou na forma de armazenamento físico. Em outras palavras, quando for necessária alguma modificação na forma como os dados são armazenados fisicamente, nenhuma alteração deve ser necessária nas aplicações que fazem uso do banco de dados. Devem também permanecer inalterados os mecanismos de consulta e manipulação de dados utilizados pelos usuários finais.

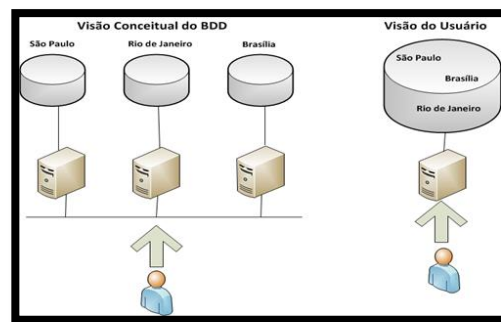
9. **Independência lógica de dados:** Mudanças nas relações e nas views provocam pouco ou nenhum impacto nas aplicações. Ou seja, as alterações nas tabelas que preservam os valores originais não devem afetar os aplicativos e recursos *ad hoc*.



10. **Independência de integridade:** As aplicações não são afetadas quando ocorrem mudanças nas regras de restrições de integridade. Deve ser possível que todas as regras de integridade sejam definidas na linguagem relacional e armazenadas no catálogo de sistema, **não** no nível de aplicação. As várias formas de integridade do banco de dados (integridade de entidade, referencial, restrição e obrigatoriedade de valores etc.) precisam ser estabelecidas dentro do catálogo do sistema ou dicionário de dados, e ser totalmente independente da lógica dos aplicativos.

11. **Independência de distribuição:** As aplicações não são logicamente afetadas quando ocorrem mudanças geográficas dos dados. Ou seja, os usuários finais **não** devem

perceber o fato de o banco de dados ser distribuído ou local. Sistemas de banco de dados distribuídos (SBDD) podem estar espalhados em diversas plataformas, interligados em rede, e podem, inclusive, estar fisicamente distantes entre si. Essa capacidade de distribuição não pode afetar a funcionalidade do sistema e dos aplicativos que fazem uso do banco de dados.



12. **Não transposição das regras:** Se um sistema possui uma linguagem de baixo nível, essa linguagem não pode ser usada para subverter as regras de integridades e as restrições definidas no nível mais alto. Ou seja, o sistema deve ser capaz de **impedir que qualquer usuário ou programador passe por cima de todos os mecanismos de segurança, das regras de integridade do banco de dados e das restrições**, utilizando algum recurso ou linguagem de baixo nível que eventualmente possam ser oferecidos pelo próprio sistema.



Agora que você já entendeu as 12 regras de Codd aplicadas a banco de dados relacionais, vamos fazer uma rápida questão sobre o assunto.



Ano: 2019 Banca: CESPE Órgão: SEFAZ-RS Prova: Auditor Assunto: Banco de Dados

Uma das regras de Codd para o modelo relacional consiste

A na dependência de dados físicos (mudança na memória e no método de acesso).

B na independência de distribuição.

C na presença de uma linguagem de programação no SGBD que promova interface com o banco de dados, com a segurança e com a atualização dos dados.

D na subversão das regras de integridade ou restrições quando utilizada uma linguagem de baixo nível.

E no não tratamento das atualizações de visões de dados.

Comentário: A alternativa B refere-se à 11ª regra de Codd:

Regra 11: Independência de Distribuição: A Distribuição de partes do SGBD em várias localidades deve ser transparente para os seus usuários. Aplicações existentes necessitam continuar operando com sucesso:

Quando uma versão distribuída do SGBD é introduzida pela primeira vez, e

Quando dados distribuídos existentes são redistribuídos em outras localidades físicas.

Gabarito: B



MAPA MENTAL – REGRAS DE CODD

Vamos agora organizar as 12 (13) regras de Codd em um mapa mental

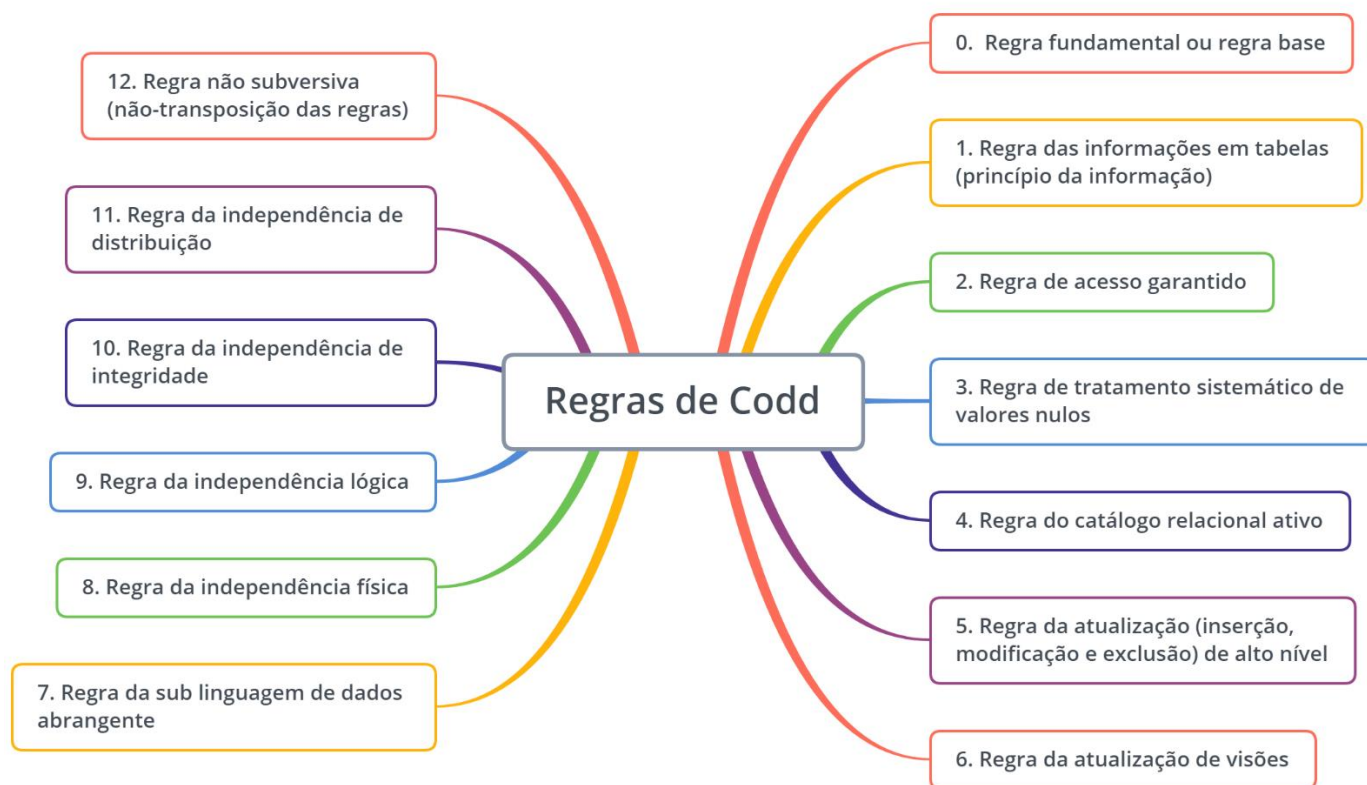


Figura 9 - Regras de Codd



ÁLGEBRA RELACIONAL

Falaremos agora sobre a **álgebra relacional**, um assunto bastante explorado em concursos públicos. Esse tema possui algumas características matemáticas que são baseadas, principalmente, na teoria de conjuntos. O conjunto básico de operações utilizadas no modelo relacional é conhecido como álgebra relacional. Navathe apresenta três motivos para considerarmos essas operações importantes:



- Fornece **alicerce formal** para o modelo relacional.
- É usado como base para **implementação** e **otimização de consultas**.
- Alguns dos seus **conceitos** são incorporados na linguagem **SQL padrão**.

Historicamente, a álgebra e o cálculo relacional foram desenvolvidos antes da linguagem SQL. O conjunto básico de operações para o modelo relacional é a álgebra relacional. Essas operações permitem que um usuário especifique as solicitações de recuperação básicas como expressões da álgebra relacional. O resultado de uma recuperação ou uma consulta é uma nova relação.

Embora a álgebra defina um conjunto de operações para o modelo relacional, o **cálculo relacional** oferece uma **linguagem declarativa** de nível mais alto para especificar consultas relacionais. A álgebra relacional, normalmente, é considerada uma parte do modelo de dados relacional. Suas operações podem ser divididas em dois grupos. Um grupo inclui conjunto de operações da teoria de conjuntos da matemática; e outro grupo, desenvolvidas especificamente para bancos de dados relacionais: as operações relacionais.

Vamos apresentar cada uma das operações e, em seguida, apresentaremos as taxonomias ou formas de classificação das operações. Tenham em mente que todas as operações são executadas sobre uma ou duas tabelas e o retorno delas é sempre outra relação. Essa característica é conhecida como **fechamento**. Veja a figura abaixo:



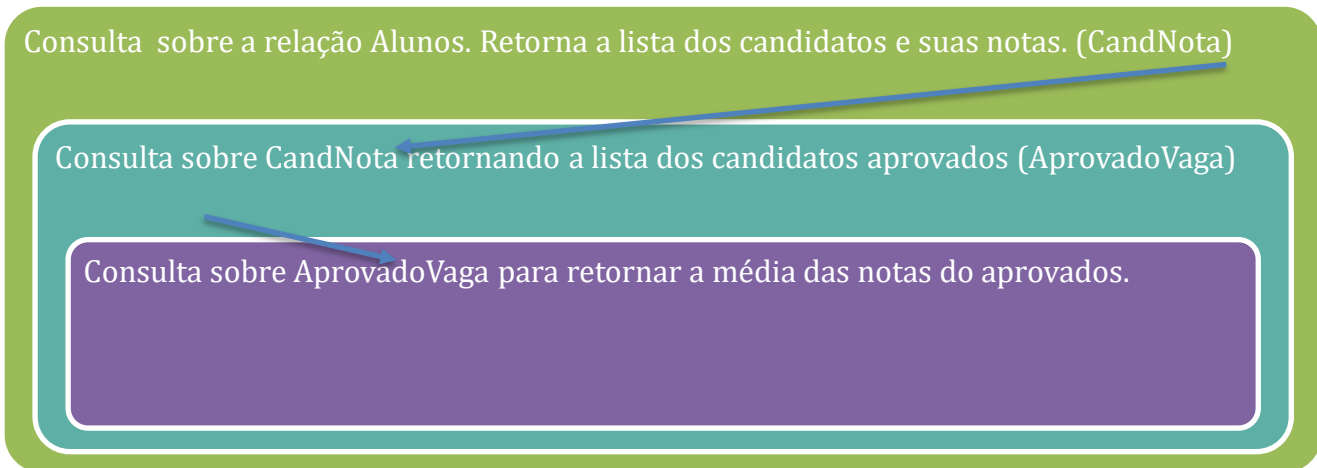


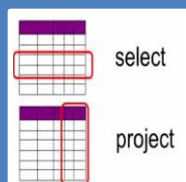
Figura 10 - Exemplo da propriedade de fechamento

Veja que, na figura acima, fazemos uma consulta na relação Alunos, retornando a lista dos candidatos e suas notas, que denominamos de CandNota. Sobre a relação CandNota, faremos uma nova pesquisa para selecionar apenas os candidatos aprovados dentro das vagas. Perceba que o resultado dessas operações será sempre outra relação que pode ser usada como entrada para outra operação. Isso é extremamente útil para o aninhamento de operações. Isso é reflexo da propriedade de fechamento.

Antes de seguir em frente a descrever cada uma das operações, gostaria de apresentar para vocês a lista de operações consideradas fundamentais da álgebra relacional. Essas 6 operações são divididas em unárias e binárias. Vou apresentar apenas a lista na figura abaixo, logo em seguida vamos tratar separadamente de cada uma destas operações.

Unárias

- Seleção
- Projeção
- Rename



Binárias

- União
- Diferença
- Produto cartesiano

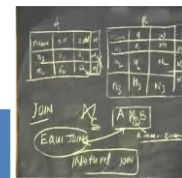


Figura 11 - Operações fundamentais da álgebra relacional.

Sobre a taxonomia, é importante lembrar que temos um conjunto de operações conhecidas como **fundamentais**. São elas: **seleção**, **projeção**, **união**, **diferença**, **produto**



cartesiano e **rename**. Essas operações possuem esse nome, pois a partir delas é possível obter o resultado das demais operações.

Outra classificação importante é a classificação das operações em binárias e unárias (ou primárias). Esse parâmetro está baseado na quantidade de relações que são utilizadas nas operações. As operações de projeção, seleção e rename são **unárias**. As operações de união, interseção, subtração, produto cartesiano, junção e divisão são **binárias**.

Operadores de conjuntos tradicionais

- **União , Interseção , Diferença e Produto cartesiano**
- Todos eles um pouco modificados para levar em conta o fato de que seus **operandos** são, especificamente, **relações** e não conjuntos arbitrários.

Operadores relacionais especiais

- **De restrição (também conhecido como seleção), projeção, junção e divisão.**

SELEÇÃO

O objetivo dessa operação, que recebe como entrada uma única tabela ou relação, é selecionar um conjunto de tuplas que satisfaçam um predicado (uma condição lógica) nos valores dos atributos. Em outras palavras, vamos extrair de uma relação um conjunto de linhas que possuem algumas restrições. Por exemplo, funcionários que possuem salário maior que 10 mil reais. Vamos procurar entender melhor o que a operação faz exatamente com outro exemplo. Vejam a tabela abaixo:

	Distribuição de petróleo no mundo (%)	Distribuição de gás natural no mundo (%)
América do Norte	3,5	5,0
América Latina	13,0	6,0
Europa	2,0	3,6
Ex-União Soviética	6,3	38,7
Oriente Médio	64,0	33,0
África	7,2	7,7
Ásia/Oceania	4,0	6,0

Queremos executar uma operação de seleção sobre esta tabela. Vamos então fazer uma consulta de quais grupos de países ou parte do mundo apresentam mais de 10% da distribuição de petróleo. Olhando para a tabela, já poderíamos trazer os valores América



Latina e Oriente Médio. Mas existe uma notação específica, definida pela álgebra relacional, para fazer essa consulta. Essa notação pode ser vista abaixo:



LISTA DE CONDIÇÕES (RELAÇÃO)

-- SELECIONA LINHAS QUE SATISFAZEM UM PREDICADO OU UMA CONDIÇÃO --

A letra grega σ (sigma) representa a operação de seleção, que é feita sobre a relação R, restringindo **as linhas de acordo com as restrições descritas pelo predicado (lista de condições)**. Sobre esse predicado, é preciso tecer alguns comentários. Primeiramente, é perfeitamente possível que ele seja uma composição de restrições sobre diferentes colunas da tabela.

Cada restrição é conhecida como **termo**. Os termos podem ser ligados por conectivos \wedge (e), \vee (ou), \neg (não). Cada termo pode ser composto por:

<atributo> operador <atributo> ou <atributo> operador <constante>
Ex: NomeCandidato = NomeAprovado || NomeAprovado = _____

Preencha com
seu nome.

No exemplo acima preciso que você imagine que NomeCandidato e NomeAprovado são atributos de uma relação. Veja que seu nome é uma constante ou um valor. Por fim, a lista de operadores é formada por =, \neq , >, \geq , < ou \leq . Percebam que você pode compor diferentes predicados com essas estruturas.

Vamos agora, que conhecemos a sintaxe e a consulta sugerida, escrever a mesma na notação correta. (Para facilitar nossa escrita, vamos chamar as colunas de REGIAO, DISTRIBUICAODEPETROLEO, DISTRIBICAODEGAS e a relação de PETROLEOREGIAO):

$\sigma_{<DISTRIBUICAODEPETROLEO \geq 10>}(PETROLEOREGIAO)$

Observe, na consulta acima, que o predicado $<DISTRIBUICAODEPETROLEO \geq 10>$ é definido sobre um atributo da tabela PETROLEOREGIAO.

Vejamos mais um exemplo da operação, desta vez vamos utilizar uma tabela com a lista de professores do Estratégia Concursos. A operação abaixo:

$\sigma_{IDADE > 35}(\text{PROFESSOR})$

Sobre a tabela:



TABELA PROFESSOR				
NOME PROFESSOR	CPF	IDADE	NOME DISCIPLINA	NATURALIDADE
DIEGO CARVALHO	111.111.111-11	21	INFORMÁTICA	DISTRITO FEDERAL
RENATO DA COSTA	222.222.222-22	54	INFORMÁTICA	RIO DE JANEIRO
RICARDO VALE	333.333.333-33	40	DIREITO CONSTITUCIONAL	MINAS GERAIS
ROSENVAL JÚNIOR	444.444.444-44	32	DIREITO AMBIENTAL	MINAS GERAIS
HERBERT ALMEIDA	555.555.555-55	19	DIREITO ADMINISTRATIVO	ESPÍRITO SANTO

Retorna os seguintes valores:

RESULTADO				
NOME PROFESSOR	CPF	IDADE	NOME DISCIPLINA	NATURALIDADE
RENATO DA COSTA	222.222.222-22	54	INFORMÁTICA	RIO DE JANEIRO
RICARDO VALE	333.333.333-33	40	DIREITO CONSTITUCIONAL	MINAS GERAIS



PROJEÇÃO

π LISTA DE ATRIBUTOS (RELAÇÃO)

-- PROJETA UMA NOVA TABELA APENAS COM OS ATRIBUTOS ESPECIFICADOS --

Agora vamos passar para a projeção. Essa operação, também, recebe como entrada uma relação. Contudo, suas restrições são feitas sobre as colunas da tabela. Essa operação deixa algumas colunas de fora do resultado, reduzindo, portanto, a quantidade de dados a serem analisados. Usando o mesmo exemplo anterior, vamos supor que queremos as informações apenas da distribuição de petróleo no mundo e das regiões. Bastaria recortar a tabela excluindo a coluna que informa a distribuição de gás.

Da mesma maneira que a operação anterior, apresentaremos agora a representação formal da projeção:

$$\pi_{A_1, A_2, \dots, A_k}(R)$$

A letra grega π (pi) representa a operação. A_1, A_2, \dots, A_k são a lista de atributos da relação R que queremos trazer como resultado. Para fazermos a consulta utilizando a nomenclatura correta, podemos escrevê-la da seguinte forma:

$$\pi_{\langle \text{REGIAO, DISTRIBUICAODEPETROLEO} \rangle}(\text{PETROLEOREGIAO})$$

Ainda sobre a essa operação, consideramos relevante saber que os resultados duplicados são removidos.

Vejamos mais um exemplo considerando a lista de professores do Estratégia Concursos. Imaginem que desejamos projetar uma nova tabela apenas com o Nome e CPF dos professores. Para tal, nós poderíamos realizar a seguinte operação algébrica: $\pi_{\text{nome professor, cpf}}(\text{PROFESSOR})$.

TABELA PROFESSOR				
NOME PROFESSOR	CPF	IDADE	NOME DISCIPLINA	NATURALIDADE
DIEGO CARVALHO	111.111.111-11	21	INFORMÁTICA	DISTRITO FEDERAL
RENATO DA COSTA	222.222.222-22	54	INFORMÁTICA	RIO DE JANEIRO
RICARDO VALE	333.333.333-33	40	DIREITO CONSTITUCIONAL	MINAS GERAIS
ROSENVAL JÚNIOR	444.444.444-44	32	DIREITO AMBIENTAL	Minas gerais
HERBERT ALMEIDA	555.555.555-55	19	DIREITO ADMINISTRATIVO	Espírito santo

resultado



NOME PROFESSOR	CPF
DIEGO CARVALHO	111.111.111- 11
RENATO DA COSTA	222.222.222- 22
RICARDO VALE	333.333.333- 33
ROSENVAL JÚNIOR	444.444.444- 44
HERBERT ALMEIDA	555.555.555- 55

UNIÃO

TABELA A TABELA B

-- RESULTA NA UNIÃO DAS LINHAS DE DUAS TABELAS COM ELIMINAÇÃO AUTOMÁTICA DE DUPLICATAS --

A operação é caracterizada por $R \cup S$, onde R e S são duas relações com a mesma quantidade de atributos, os quais operam sobre o mesmo domínio. O resultado contém as tuplas que estão em R, S ou ambas. Suponha que R seja a relação professor e S a relação aluno, se quisermos fazer a união das duas relações temos que:

TABELA PROFESSOR				TABELA ALUNO		
NOME	CPF	CÓDIGO		NOME	CPF	CÓDIGO
DIEGO CARVALHO	000.000.000-00	101		ROMÁRIO	555.555.555-55	101
RENATO DA COSTA	111.111.111-11	101		ROBERTO CARLOS	666.666.666-66	101
RICARDO VALE	222.222.222-22	102		RONALDO FOFO	777.777.777-77	102
ROSENVAL JÚNIOR	333.333.333-33	103		RIVALDO	888.888.888-88	103
HERBERT ALMEIDA	444.444.444-44	104		RONALDO GAÚCHO	999.999.999-99	104

RESULTADO		
NOME	CPF	CÓDIGO
DIEGO CARVALHO	000.000.000-00	101
RENATO DA COSTA	111.111.111-11	101
RICARDO VALE	222.222.222-22	102
ROSENVAL JÚNIOR	333.333.333-33	103
HERBERT ALMEIDA	444.444.444-44	104
ROMÁRIO	555.555.555-55	101
ROBERTO CARLOS	666.666.666-66	101
RONALDO FOFO	777.777.777-77	102
RIVALDO	888.888.888-88	103
RONALDO GAÚCHO	999.999.999-99	104



Perceba que a operação de união vai juntar as linhas das tabelas verticalmente, colocando uma sobre a outra. Essa operação somente pode ser realizada se as tabelas forem união compatíveis, isto é, possuírem a mesma estrutura. Nesta situação, o domínio de cada um dos atributos deve ser igual.

INTERSECÇÃO

TABELA A \cap TABELA B

-- RESULTA EM UMA NOVA TABELA QUE CONTÉM OS ELEMENTOS EM COMUM ÀS DUAS TABELAS SEM REPETIÇÕES --

A intersecção é representada por $R \cap S$, traz como resultado a relação que contém as tuplas que estão em ambas R e S. Ou seja, trata-se de uma operação que produz como resultado uma tabela que contém, sem repetições, todos os elementos que são comuns às duas tabelas fornecidas como operandos. Vamos agora, mostrar um exemplo da operação de intersecção entre as tabelas professor escolar e professor universitário.

TABELA PROFESSOR ESCOLAR		
NOME	CPF	DT NASCIMENTO
DIEGO CARVALHO	111.111.111-11	12/10/1988
RENATO DA COSTA	222.222.222-22	11/04/1961
RICARDO VALE	333.333.333-33	17/07/1979
ROSENVAL JÚNIOR	444.444.444-44	01/12/1983
HERBERT ALMEIDA	555.555.555-55	28/02/1977



TABELA PROFESSOR UNIVERSITÁRIO		
NOME	CPF	DT NASCIMENTO
DIEGO CARVALHO	111.111.111-11	12/10/1988
MARCOS GIRÃO	666.666.666-66	01/08/1968
DÉCIO TERROR	777.777.777-77	27/06/1976
RENATO DA COSTA	999.999.999-99	11/04/1961
GUILHERME NEVES	888.888.888-88	11/04/1961

RESULTADO		
NOME	CPF	DT NASCIMENTO
DIEGO CARVALHO	111.111.111-11	12/10/1988

DIFERENÇA

TABELA A - TABELA B

-- RESULTA EM UMA NOVA TABELA QUE CONTÉM AS LINHAS PRESENTES NA TABELA A E AUSENTES NA TABELA B --

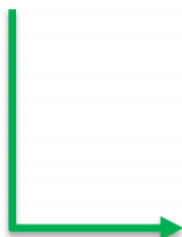
Quando pensamos na relação $R - S$, a subtração ou diferença apresenta como resultado outra relação que contém as tuplas que estão em R e que não estão em S. Veja que, ao contrário das operações de união e intersecção listadas acima, a diferença não é comutativa, ou seja, $R - S \neq S - R$. No caso do exemplo abaixo, observem que o resultado



trará as linhas que estão na Tabela **PROFESSOR ESCOLAR** que não estão na Tabela **PROFESSOR UNIVERSITÁRIO**.

TABELA PROFESSOR ESCOLAR		
NOME	CPF	DT NASCIMENTO
DIEGO CARVALHO	111.111.111-11	12/10/1988
RENATO DA COSTA	222.222.222-22	11/04/1961
RICARDO VALE	333.333.333-33	17/07/1979
ROSENVAL JÚNIOR	444.444.444-44	01/12/1983
HERBERT ALMEIDA	555.555.555-55	28/02/1977

TABELA PROFESSOR UNIVERSITÁRIO		
NOME	CPF	DT NASCIMENTO
DIEGO CARVALHO	111.111.111-11	12/10/1988
MARCOS GIRÃO	666.666.666-66	01/08/1968
DÉCIO TERROR	777.777.777-77	27/06/1976
RENATO DA COSTA	222.222.222-22	11/04/1961
GUILHERME NEVES	888.888.888-88	11/04/1985



RESULTADO		
NOME	CPF	DT NASCIMENTO
RICARDO VALE	333.333.333-33	17/07/1979
ROSENVAL JÚNIOR	444.444.444-44	01/12/1983
HERBERT ALMEIDA	555.555.555-55	28/02/1977

PRODUTO CARTESIANO

(Relação A) X (Relação B)

-- RESULTA EM UMA NOVA TABELA COM TODAS AS COMBINAÇÕES DE LINHAS DE AMBAS AS RELAÇÕES --

Permite combinar informações de duas relações, fazendo uma junção de todas as linhas, a primeira com todas as linhas da segunda. O produto cartesiano entre duas tabelas ou relações gera uma nova relação com **a quantidade de tuplas igual ao produto da quantidade de tuplas de cada uma das relações**. A nova relação possui todos os atributos que compõem cada uma das tabelas, fazendo parte da operação da seguinte forma:



TABELA PROFESSOR	
NOME PROFESSOR	CPF
DIEGO CARVALHO	111.111.111-11
RENATO DA COSTA	222.222.222-22
RICARDO VALE	333.333.333-33
ROSENVAL JÚNIOR	444.444.444-44
HERBERT ALMEIDA	555.555.555-55

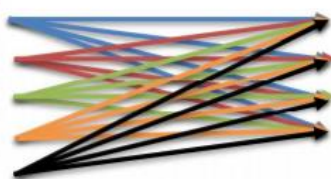



TABELA DISCIPLINA	
NOME DISCIPLINA	CÓDIGO
INFORMÁTICA	101
DIREITO CONSTITUCIONAL	102
DIREITO AMBIENTAL	103
DIREITO ADMINISTRATIVO	104

RESULTADO			
NOME PROFESSOR	CPF	NOME DISCIPLINA	CÓDIGO
DIEGO CARVALHO	111.111.111-11	INFORMÁTICA	101
RENATO DA COSTA	222.222.222-22	INFORMÁTICA	101
RICARDO VALE	333.333.333-33	INFORMÁTICA	101
ROSENVAL JÚNIOR	444.444.444-44	INFORMÁTICA	101
HERBERT ALMEIDA	555.555.555-55	INFORMÁTICA	101
DIEGO CARVALHO	111.111.111-11	DIREITO CONSTITUCIONAL	102
RENATO DA COSTA	222.222.222-22	DIREITO CONSTITUCIONAL	102
RICARDO VALE	333.333.333-33	DIREITO CONSTITUCIONAL	102
ROSENVAL JÚNIOR	444.444.444-44	DIREITO CONSTITUCIONAL	102
HERBERT ALMEIDA	555.555.555-55	DIREITO CONSTITUCIONAL	102
DIEGO CARVALHO	111.111.111-11	DIREITO AMBIENTAL	103
RENATO DA COSTA	222.222.222-22	DIREITO AMBIENTAL	103
RICARDO VALE	333.333.333-33	DIREITO AMBIENTAL	103
ROSENVAL JÚNIOR	444.444.444-44	DIREITO AMBIENTAL	103
HERBERT ALMEIDA	555.555.555-55	DIREITO AMBIENTAL	103
DIEGO CARVALHO	111.111.111-11	DIREITO ADMINISTRATIVO	104
RENATO DA COSTA	222.222.222-22	DIREITO ADMINISTRATIVO	104
RICARDO VALE	333.333.333-33	DIREITO ADMINISTRATIVO	104
ROSENVAL JÚNIOR	444.444.444-44	DIREITO ADMINISTRATIVO	104
HERBERT ALMEIDA	555.555.555-55	DIREITO ADMINISTRATIVO	104

JUNÇÃO

RELAÇÃO A CONDIÇÃO RELACÃO B

-- RESULTA EM UMA NOVA TABELA COM TODAS AS COMBINAÇÕES DE LINHAS QUE SATISFAZEM ALGUMA CONDIÇÃO --

A junção é uma conexão entre duas tabelas na qual elas são **mescladas de acordo com um ou mais campos em comum**. A junção pode ser considerada um produto cartesiano seguido por uma seleção. O símbolo  representa uma junção. Ao executar uma junção, ela deve satisfazer a um predicado θ e deve existir pelo menos um campo nas duas tabelas que operem sobre o mesmo domínio. Esse campo é usado para decidir quais linhas da



primeira tabela devem se relacionar com cada uma das linhas da segunda tabela. Geralmente, exige-se que os valores dos atributos sejam iguais.

Se desejarmos apresentar – em uma única tabela – as linhas da Tabela PROFESSOR e da Tabela DISCIPLINA cujo CÓDIGO (PROFESSOR) seja igual a CÓDIGO (DISCIPLINA), temos que: **PROFESSOR** ⋈ CÓDIGO = CÓDIGO **DISCIPLINA**.

TABELA PROFESSOR					TABELA DISCIPLINA	
NOME PROFESSOR	CPF	CÓDIGO			NOME DISCIPLINA	CÓDIGO
DIEGO CARVALHO	111.111.111-11	101			INFORMÁTICA	101
RENATO DA COSTA	222.222.222-22	101			DIREITO CONSTITUCIONAL	102
RICARDO VALE	333.333.333-33	102			DIREITO AMBIENTAL	103
ROSENVAL JÚNIOR	444.444.444-44	103			DIREITO ADMINISTRATIVO	104
HERBERT ALMEIDA	555.555.555-55	104				

RESULTADO			
NOME PROFESSOR	CPF	CÓDIGO	NOME DISCIPLINA
DIEGO CARVALHO	111.111.111-11	101	INFORMÁTICA
RENATO DA COSTA	222.222.222-22	101	INFORMÁTICA
RICARDO VALE	333.333.333-33	102	DIREITO CONSTITUCIONAL
ROSENVAL JÚNIOR	444.444.444-44	103	DIREITO AMBIENTAL
HERBERT ALMEIDA	555.555.555-55	104	DIREITO ADMINISTRATIVO

Perceba que a condição de igualdade foi imposta no exemplo acima, onde o código deveria ser igual nas duas tabelas para que a linha aparecesse no resultado. Contudo, existem variações da junção que não condiram apenas a igualdade dos atributos conforme veremos abaixo.

Natural join (⋈): Junção na qual θ é uma igualdade predefinida entre todos os atributos de **mesmo nome** presentes em duas relações R1 e R2 (atributos de junção). Estes atributos só aparecem uma vez no resultado.

Junções externas ou outer joins. Junção na qual **as tuplas** de uma ou de ambas as relações, que não são combinadas (ou seja, não existe valor correspondente na outra relação que satisfaça a condição de junção), mesmo assim são preservadas no resultado. Possui basicamente três variações. Logo em seguida, apresentamos um exemplo.

Left outer Join: tuplas da relação à esquerda são preservadas.

Notação: $\bowtie \leftarrow$

Right outer join: tuplas da relação à direita são preservadas

Notação: $\bowtie \rightarrow$

Full outer join: tuplas da relação à direita e à esquerda são preservadas

Notação: $\bowtie \longleftrightarrow$



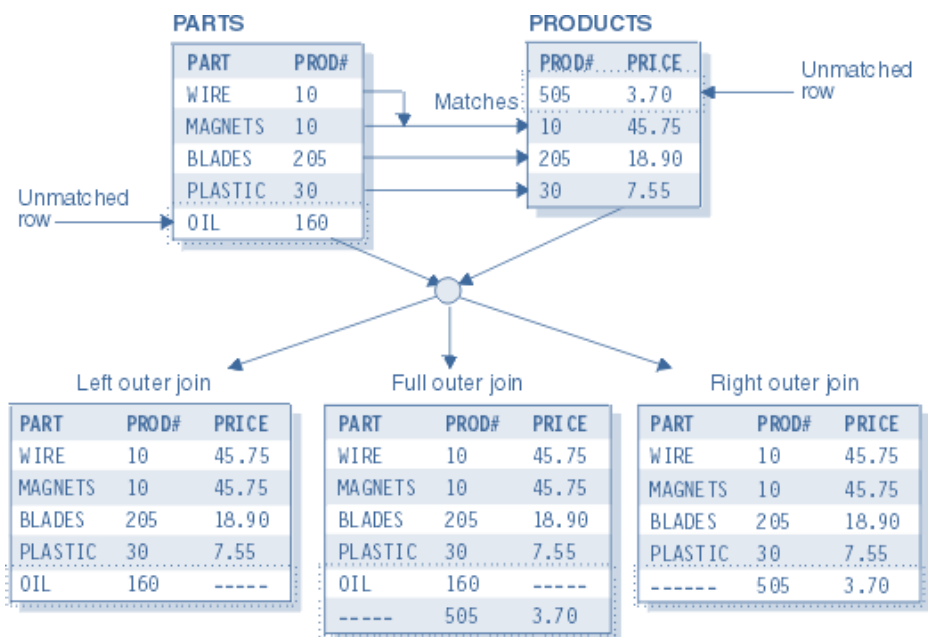
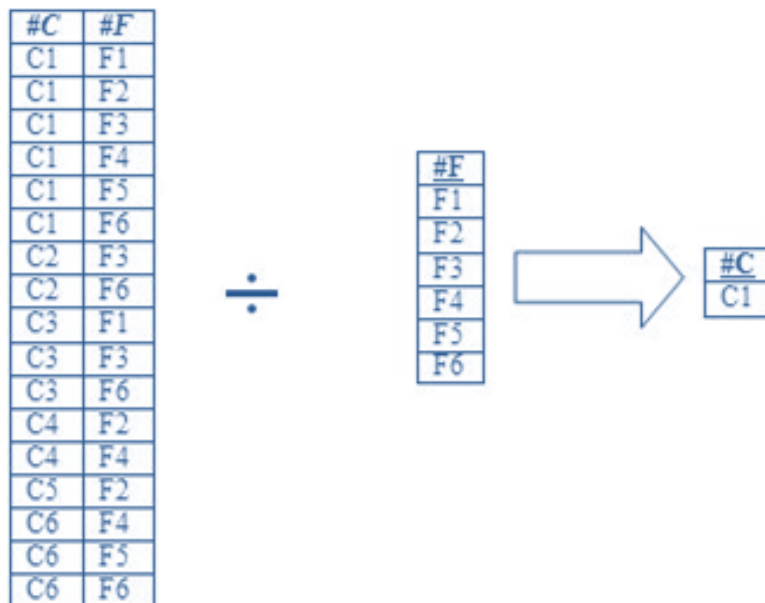


Figura 12 - Exemplo de left, full e right outer join.

DIVISÃO

A operação de divisão é usada nas consultas nas quais se emprega a frase: “para todos”. Seu resultado será composto, basicamente, pelos elementos da primeira tabela que se relacionem com todos os elementos da segunda tabela. Vejam o exemplo abaixo:



BANCA: FCC ANO: 2012 ÓRGÃO: TJ-RJ PROVA: ANALISTA JUDICIÁRIO - ANALISTA DE SISTEMAS



Considere a seguinte tabela de um banco de dados.

$TAB_FUNC = \{COD_FUNC, NOME, COD_DEP, SAL\}$

Uma expressão da álgebra relacional representando a tabela formada pelos códigos (COD_FUNC) e nomes (NOME) dos funcionários que ganham salário (SAL) entre 1000 e 3000 reais e trabalham no departamento de código (COD_DEP) 3 é

- A $\pi_{COD_FUNC, NOME}(\sigma_{COD_DEP=3 \wedge SAL \geq 1000 \wedge SAL \leq 3000}(TAB_FUNC))$
- B $\pi(\sigma_{SAL \geq 1000 \wedge SAL \leq 3000}(TAB_FUNC))$
- C $\sigma_{COD_DEP}(\pi_{SAL \geq 1000 \wedge SAL \leq 3000}(TAB_FUNC))$
- D $\pi(\sigma_{SAL \geq 1000 \wedge SAL \leq 3000 \wedge COD_DEP=3}(COD_FUNC, NOME))$
- E $\sigma_{NOME, COD_FUNC}(\pi_{COD_DEP=3 \wedge SAL \geq 1000 \wedge SAL \leq 3000}(TAB_FUNC))$

Comentário: Uma das opções para resolver essa questão é construir o código SQL da consulta:

```
SELECT COD_FUNC, NOME FROM TAB_FUNC WHERE  
SAL >= 1000 AND SAL <= 3000 AND COD_DEP = 3;
```

Agora vamos transformar essa consulta para a álgebra relacional. Vamos fazer uma projeção dos atributos COD_FUNC e NOME sobre o resultado de uma seleção com as restrições iguais às do select. Assim, temos como resultado:

$$\pi_{\langle COD_FUN, NOME \rangle}(\sigma_{\langle SAL \geq 1000 \wedge SAL \leq 3000 \wedge COD_DEP = 3 \rangle}(TAB_FUN))$$

Vejam que esse resultado está presente na alternativa A com algumas modificações na ordem das restrições impostas pelo comando de seleção.

Gabarito: A.

Ok! Vamos passar agora para o nosso tradicional mapa mental! Antes, porém, gostaria que você se lembrasse das 8 operações principais da álgebra relacional que foram definidas na primeira versão dos operadores.



TABELA COM O RESUMO DAS OPERAÇÕES DA ÁLGEBRA RELACIONAL.

OPERAÇÃO	FINALIDADE	NOTAÇÃO
SELEÇÃO	Seleciona todas as tuplas que satisfazem a condição de seleção de uma relação R .	$\sigma_{\langle \text{condição seleção} \rangle}(R)$
PROJEÇÃO	Produz uma nova relação com apenas alguns dos atributos de R , e remove tuplas duplicadas.	$\pi_{\langle \text{lista atributos} \rangle}(R)$
JUNÇÃO THETA	Produz todas as combinações de tuplas de R_1 e R_2 que satisfazem a condição de junção.	$R_1 \bowtie_{\langle \text{condição junção} \rangle} R_2$
EQUIJUNÇÃO	Produz todas as combinações de tuplas de R_1 e R_2 que satisfazem uma condição de junção apenas com comparações de igualdade.	$R_1 \bowtie_{\langle \text{condição junção} \rangle} R_2$, OR $R_1 \bowtie_{(\langle \text{atributos junção 1} \rangle), (\langle \text{atributos junção 2} \rangle)} R_2$
JUNÇÃO NATURAL	O mesmo que EQUIJOIN, exceto que atributos de junção de R_2 não são incluídos na relação resultante; se os atributos de junção tiverem os mesmos nomes, eles nem sequer precisam ser especificados.	$R_1 \star_{\langle \text{condição junção} \rangle} R_2$, OR $R_1 \star_{(\langle \text{atributos junção 1} \rangle), (\langle \text{atributos junção 2} \rangle)} R_2$ OR $R_1 \star R_2$
UNIÃO	Produz uma relação que inclui todas as tuplas em R_1 ou R_2 ou tanto R_1 quanto R_2 ; R_1 e R_2 precisam ser compatíveis na união.	$R_1 \cup R_2$
INTERSECÇÃO	Produz uma relação que inclui todas as tuplas em R_1 e R_2 ; R_1 e R_2 precisam ser compatíveis na união.	$R_1 \cap R_2$
DIFERENÇA	Produz uma relação que inclui todas as tuplas em R_1 que não estão em R_2 ; R_1 e R_2 precisam ser compatíveis na união.	$R_1 - R_2$
PRODUTO CARTESIANO	Produz uma relação que tem os atributos de R_1 e R_2 e inclui como tuplas todas as possíveis combinações de tuplas de R_1 e R_2 .	$R_1 \times R_2$
DIVISÃO	Produz uma relação $R(X)$ que inclui todas as tuplas $t[X]$ em $R_1(Z)$ que aparecem em R_1 em combinação com toda tupla de $R_2(Y)$, onde $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$

Figura 13 - Resumo das operações da álgebra relacional.

Uma sugestão de tarefa é que você complete o mapa mental da página seguinte com a notação e palavras-chave que ajudarão você a lembrar do significado de cada operação da álgebra relacional.



MAPA MENTAL – ÁLGEBRA RELACIONAL

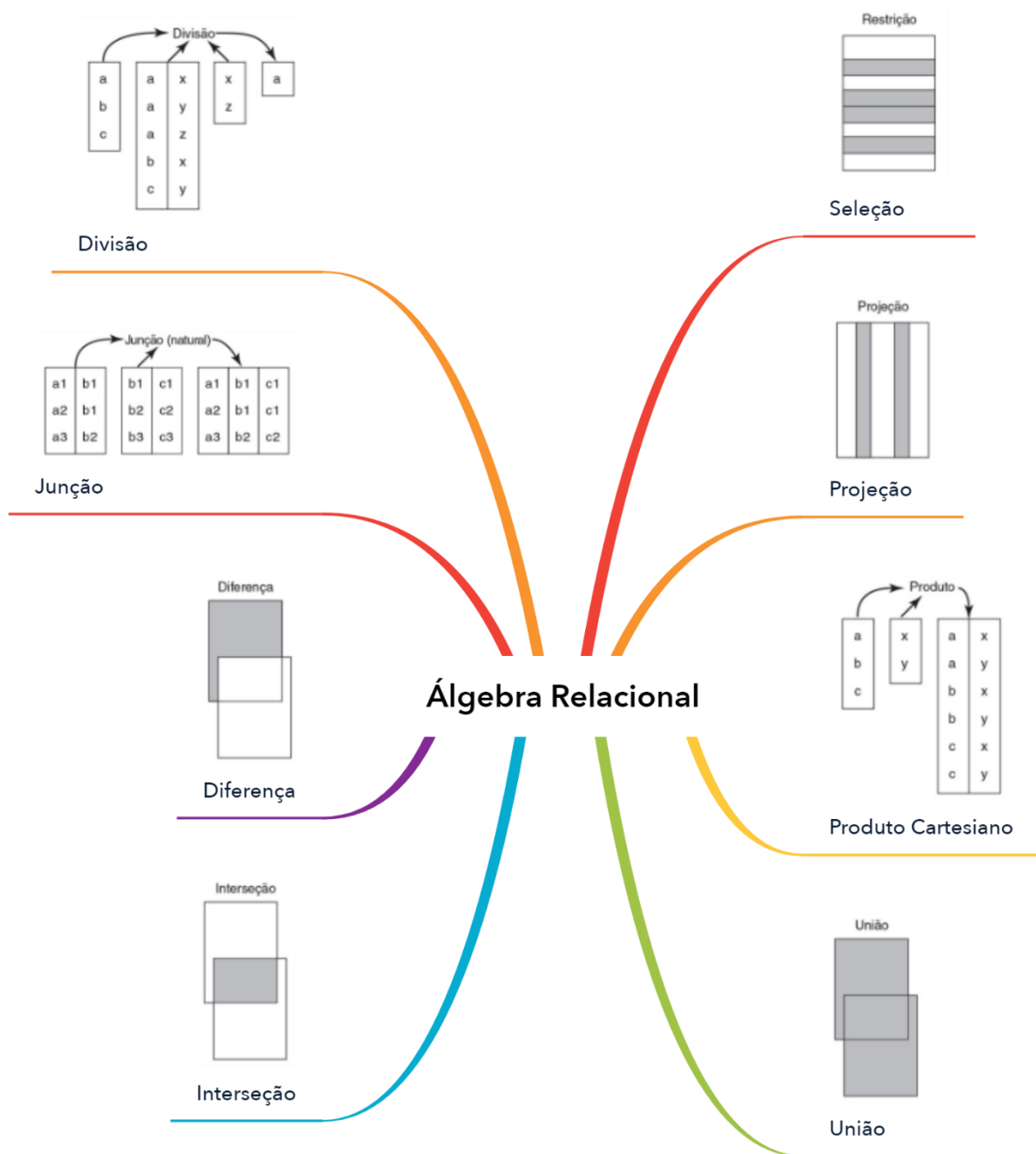


Figura 14 - Mapa mental das operações da álgebra relacional





NORMALIZAÇÃO

INTRODUÇÃO

Normalizar em sentido amplo significa voltar à ordem. Mas que ordem seria essa que precisa ser mantida quando executando um projeto de banco de dados? Um bom projeto de banco de dados garante ao usuário a possibilidade de modificar o conteúdo do banco de dados sem causar efeitos colaterais inesperados ou anomalias de atualização (ou modificação). Uma **anomalia de modificação** é um efeito colateral inesperado que ocorre quando se alteram os dados de uma tabela com redundâncias excessivas.

Observe a tabela abaixo, perceba que estamos observando que o cliente Thiago está comprando alguns produtos para começar a estudar:

TABELA VENDAS

Cod_Compra	Nome	Endereço	CPF	Produto
1	Thiago	SQSW 302	999999999	Caneta
2	Thiago	SQSW 302	999999999	Livro
3	Thiago	SQSW 302	999999999	Borracha

Perceba que os dados realçados em amarelo são repetidos ou redundantes em cada uma das linhas. Agora imagine que eu resolva atualizar o endereço de Thiago. Eu acessei a primeira linha da tabela e alterei a informação conforme desejava. Veja que o endereço de Thiago mudou para **SQSW 304**

TABELA VENDAS

Cod_Compra	Nome	Endereço	CPF	Produto
1	Thiago	SQSW 304	999999999	Caneta
2	Thiago	SQSW 302	999999999	Livro
3	Thiago	SQSW 302	999999999	Borracha



Agora, o cliente Thiago, associado ao CPF 999999999, possui 2 endereços diferentes. Veja que a causa para esse acontecimento foi justamente a redundância dos dados. Se o endereço de Thiago estivesse separado em outra relação essa anomalia não teria se formado. Vejamos como ficaria:

TABELA VENDAS

Cod_Compra	CPF	Produto
1	999999999	Caneta
2	999999999	Livro
3	999999999	Borracha

TABELA CLIENTE

Nome	Endereço	CPF
Thiago	SQSW 304	999999999

Agora temos um conjunto de tabelas que possuem apenas um registro para o endereço de Thiago. Desta forma, se atualizarmos a primeira linha da TABELA CLIENTE, não geramos uma anomalia de atualização e ainda por cima reduzimos a redundância dos dados.

Show hein!? 😊

CONCEITOS BÁSICOS

O **processo de normalização**, como foi inicialmente **proposto por Codd** (1972), ele sujeita um esquema de relação a uma série de testes para certifica-se de que ele satisfaça certa **forma normal**. De uma forma mais simples, o processo observa o modelo de dados e faz algumas checagens, casos as regras não sejam verificadas é necessário agir sobre o modelo para **organizar melhor os dados**. Vejamos uma definição forma de normalização:

Definição: A **normalização de dados** é uma **técnica de decomposição** utilizada no projeto de banco de dados com objetivo de prover um armazenamento consistente, evitando **redundância de dados** e **anomalias** de atualização.



(Ministério da Economia – Especialista em Ciência de Dados - 2020) Julgue os itens a seguir, a respeito de banco de dados relacionais.

O processo de normalização de dados consiste em encontrar informações que atinjam um plano de normalização com as informações constantes nas tuplas adjacentes.

Comentários: O processo de normalização visa reduzir a redundância e as anomalias de atualização. Logo, temos uma alternativa errada.

Gabarito: ERRADO.

Observe que a definição acima fala em decomposição. Desta forma, quando aplicamos a normalização sobre um conjunto de tabelas, é normal que surjam novas colunas ou relações para que os dados possam ser mais bem distribuídos pelas tabelas.

Inicialmente, Codd propôs três formas normais: 1ª, 2ª e 3ª. Todas baseadas nas **dependências funcionais** entre os atributos de uma relação. Uma nova forma normal foi proposta por Boyce-Codd, que é mais forte do que 3ª FN. Veja que estamos falando de um conceito novo, logo precisamos explicá-lo, vamos então aproveitar para apresentar outros conceitos que nos ajudam a entender melhor o assunto.

Para entender as formas normais é preciso, antes de qualquer coisa, entender alguns conceitos. Para facilitar o seu entendimento vamos defini-lo e explicá-los em uma linguagem simples e objetiva. Vejamos...

DEPENDÊNCIA FUNCIONAL

A dependência funcional (DF) é **uma restrição de duas ou mais colunas de uma tabela**. Considerando X e Y colunas de uma tabela, podemos dizer que **X determina Y** ($X \rightarrow Y$) se existe no máximo **um valor de Y para cada valor de X**. Por exemplo, o número do cadastro de pessoa física determina a cidade (**CPFAluno** \rightarrow CidadeAluno) na tabela do banco de dados de uma universidade, se existir no máximo um valor de cidade para cada número do cadastro de pessoa física. Vejamos alguns valores:

CPFAluno	CidadeAluno	Turma
000000000000	Recife	A
111111111111	Brasília	B
222222222222	Fortaleza	C
333333333333	Ceilândia	D
444444444444	João Pessoa	E

Na DF **CPFAluno** \rightarrow CidadeAluno, a coluna que aparece à esquerda de uma DF é denominada **determinante**, ou lado esquerdo (*left-hand side* = lado da mão esquerda). Nesse exemplo, CPFAluno é um **determinante**. Também é possível pensar nas dependências funcionais como **identificadoras de potenciais chaves candidatas**. Ao



declarar que $X \rightarrow Y$, se X e Y forem colocados juntos em uma tabela sem outras colunas, **X é uma chave candidata**.

Todo determinante (lado esquerdo) é chave candidata se for colocado com outras colunas determinadas por ele. Por exemplo, se CPFAluno, CidadeAluno e Turma forem colocados juntos em uma tabela e CPFAluno \rightarrow CidadeAluno e CPFAluno \rightarrow Turma, então **CPFAluno é uma chave candidata**. Se não houver outras chaves candidatas, um determinante se tornará a chave primária se ele não permitir valores nulos.

Formalmente, uma dependência funcional, denotada por $X \rightarrow Y$ entre dois conjuntos de atributos X e Y, que são subconjunto de R, especifica uma restrição nas possíveis tuplas que formam um estado da relação r de R. A restrição é que, para quaisquer duas tuplas t_1 e t_2 em r que tenha $t_1[X] = t_2[X]$, elas também têm de ter $t_1[Y] = t_2[Y]$.

Vamos olhá-la mais uma vez para a tabela do início da aula. Perceba que nas linhas o atributo nome possui o mesmo valor, logo, se Nome \rightarrow Endereço, então a coluna endereço terá o mesmo valor para as respectivas linhas. Formalmente, temos que:

$$t_1[\text{Nome}] = t_2[\text{Nome}] \rightarrow t_1[\text{Endereço}] = t_2[\text{Endereço}].$$

Cod_Compra	Nome	Endereço	CPF	Produto
1	Thiago	SQSW 302	999999999	Caneta
2	Thiago	SQSW 302	999999999	Livro
3	Thiago	SQSW 302	999999999	Borracha

Mais uma vez: a questão aqui é perceber que se você escolher um determinado valor para o atributo X da relação (sabendo que $X \rightarrow Y$) é possível saber o valor do atributo Y. Na definição formal é dito que se duas tuplas tiverem o mesmo valor para X, então elas também terão o mesmo valor de Y. É importante lembrar também que quando $X \rightarrow Y$ (X determina Y), X é chamado de **determinante** e Y de **determinado**.

Uma dependência funcional pode ser considerada **trivial**, neste caso se $X \rightarrow Y$ e X contém (⊇) Y. Para entender isso basta supor um conjunto de atributos (A, B, C, D, E) e o fato de A, B, **C \rightarrow C** faz desta DF uma **dependência trivial**. Por outro lado, temos a possibilidade da DF ser **não trivial** se $X \rightarrow Y$ e X não contém Y. Usando os mesmos atributos acima, uma DF A, B \rightarrow D pode ser considerada um exemplo de DF não trivial.



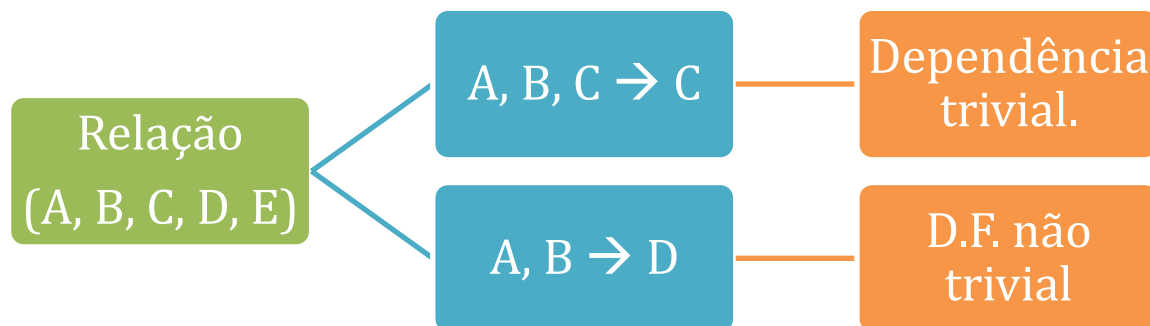


Figura 15 - Dependência trivial e não trivial

CHAVE, SUPERCHAVE, CHAVE PRIMÁRIA E CHAVE CANDIDATA

Vamos aproveitar que estamos falando de chaves para relembrar seu conceito. Existem alguns conceitos relacionados a chave. O primeiro deles é o de **superchave**. Uma definição formal afirma que uma superchave de um esquema de relação $R = \{A_1, A_2, \dots, A_n\}$ é um conjunto de atributos $S \subset R$ (S está contido em R) que contenha a propriedade na qual não haverá duas tuplas t_1 e t_2 , em qualquer estado válido da relação r de R , cuja $t_1[S] = t_2[S]$. Em outras palavras, **uma superchave é um conjunto de atributos que tem a característica de restringir o conjunto de tuplas de uma relação a apenas uma linha.**

Quando olhamos para um conjunto de atributos em uma tabela que não se repete em nenhuma das linhas da tabela podemos considerá-los uma **superchave**.

Já entendemos o conceito de superchave. Essa superchave pode ser considerada apenas uma chave. Neste caso, a chave é defendida como uma **superchave mínima** (vou chamar ela de K), onde qualquer remoção de atributo de K fará com que K deixe de ser superchave da relação. Se um esquema tiver mais de uma chave, cada uma delas é chamada de **chave candidata**. Entre as chaves candidatas uma delas é escolhida para ser a chave da relação e é denominada de **chave primária**. As demais são renegadas e são denominadas **chaves secundárias**.

Entender o conceito de chave é o primeiro passo para o entendimento das formas normais. De posse deste conhecimento, é possível definir o conceito de **atributo primário**, que nada mais é do que um atributo membro (que faz parte) de alguma **chave candidata** da relação (R). Por sua vez, de forma bem intuitiva, um **atributo não primário** é todo aquele que não for um atributo primário!



ANO: 2015 ÓRGÃO: TJDFT PROVA: PROGRAMAÇÃO DE SISTEMAS

Julgue os itens seguintes a respeito de banco de dados.

[61] Em uma tabela de um banco de dados relacional, se uma restrição de chave primária for definida como composta de mais de uma coluna, os seus valores

poderão ser duplicados em uma coluna; no entanto, cada combinação de valores de todas as colunas na definição da restrição de chave primária deve ser exclusiva.

Comentário: Questão interessante pois nos permite fazer um rápido comentário sobre chaves. Uma **chave** identifica unicamente uma linha de uma relação. Toda relação pode ter vários conjuntos de atributos que podem ser escolhidos como **chave primária**. Cada uma dessas opções que se caracterizam por ser uma **superchave mínima**, ou seja, não é possível retirar nenhum atributo sem que o conjunto perca a propriedade de ser chave da relação, é denominada **chave candidata**. A chave escolhida para ser a chave da relação é denominada chave primária. Ela pode ser **composta** por **um ou mais atributos**.

A questão pede para analisarmos uma característica de chaves compostas por mais de um atributo. Percebam que a unicidade dos valores deve considerar o conjunto dos atributos e não um atributo individualmente. Sendo assim a alternativa encontra-se **correta**.

Gabarito: C.

Com esses conceitos já é possível definir as três primeiras formas normais, bem como a forma normal de Boyce-Codd.

PRIMEIRA FORMA NORMAL (1FN)

Essa forma normal é considerada uma parte da definição de relação no modelo relacional básico. Sua definição prevê que **todos os atributos de uma relação devem ter seus valores definidos sobre domínios atômicos ou indivisíveis**. Em outras palavras, os campos de uma tabela não devem ser **compostos** ou **multivalorados**. Veja abaixo uma tabela que não está na 1FN para duas relações sendo transformada em duas relações.

Lembre-se que um atributo composto possui mais de valor de tipos diferentes associados a uma coluna da tabela. No nosso exemplo, vamos usar o atributo Endereço com valores de Rua, Número, Bairro e Cep. Já o atributo multivalorado pode ter 0, 1 ou vários valores associados ao mesmo tipo de dados. Neste caso, selecionamos o atributo telefone como exemplo.

Código_Aluno	Nome	Telefones	Endereço
A001	Fabício Ribeiro	99564-9453 98432-1234	Rua 17 de Julho, 98, Morumbi, 12635-965
B001	Carlos Normando		Rua Águas de Março, 16, Rio de Janeiro, 54532-098
C001	Emerson Pimentel	92834-5697 91283-4309	Praça Ramos 15, Liberdade, 66858-633

Você consegue perceber que temos dois atributos acima que **não** são atômicos. Claramente, podemos dividir os valores presentes nas colunas Telefones e Endereço. Vamos trabalhar cada um dos casos separadamente, começando pela coluna Telefones. Para acabar com os atributos multivalores precisamos criar uma outra tabela que vai associar cada um dos telefones ao seu respectivo aluno. Vejamos:



Código_Aluno	Telefone
A001	99564-9453
A001	98432-1234
C001	92834-5697
C001	91283-4309

Perceba que a chave da tabela acima é composta pelos atributos Código_Aluno e Telefone.

Agora, precisamos voltar a nossa atenção para a relação original e separar os diversos atributos atômicos da coluna Endereço.

Código_Aluno	Nome	Logradouro	Número	Bairro	Cep
A001	Fabício Ribeiro	Rua 17 de Julho	98	Morumbi	12635-965
B001	Carlos Normando	Rua Águas de Março	16	Rio de Janeiro	54532-098
C001	Emerson Pimentel	Praça Ramos	15	Liberdade	66858-633

Agora sim!! Temos uma tabela com todos os seus atributos associados a domínios indivisíveis ou atômicos. Veja que, para resolver o problema de atributos compostos, separamos os valores em várias colunas. Já, para resolver o problema de atributos multivalorados, criamos uma tabela com a chave da relação original e uma coluna atômica que representa os diversos/múltiplos valores em linhas distintas.

Vamos aproveitar para expor algumas definições de diversos autores sobre a primeira forma normal:

Primeira forma normal

- Uma relação R existe na primeira forma normal (1FN) se, e somente se, todos os **domínios subjacentes** contiverem apenas **valores atômicos**.
- No modelo relacional, um domínio é atômico se os elementos do domínio são considerados **unidades indivisíveis**. Um esquema de relação R está na primeira forma normal se **todos os atributos de R são atômicos**.
- A primeira forma normal afirma que o **domínio** de um atributo deve **incluir apenas valores atômicos (simples e indivisíveis)** e que o valor de qualquer atributo em uma tupla deve ser um único valor do domínio desse atributo.
- A primeira forma normal **evita** as chamadas relações aninhadas, essas relações contêm vários atributos em uma única coluna e **não** são permitidas no modelo relacional.

SEGUNDA FORMA NORMAL (2FN)

A segunda forma normal visa resolver um problema de dependência parcial. Uma relação com uma chave primária (ou candidata) composta onde um dos atributos determina



isoladamente outro atributo da relação. Vamos tentar dar um exemplo para fixarmos melhor os conceitos. Veja a tabela abaixo:

Cod Compra	CPF	Nome	Endereço	Produto
1	999999999	Thiago	SQSW 302	Caneta
2	999999999	Thiago	SQSW 302	Livro
3	888888888	Flávia	SQSW 304	Borracha

Na relação anterior, os atributos Cod_Compra e CPF formam a chave primária composta da tabela vendas. Contudo, é possível perceber que CPF, isoladamente, determina os atributos Nome e Endereço. Veja que, o exemplo que utilizamos anteriormente na nossa aula, possui problemas com a segunda forma normal. Essa dependência parcial precisa ser removida da relação. Para isso, o CPF se mantém como atributo na relação original e uma nova relação é criada para absolver os atributos por ele determinado. Assim temos:

Cod Compra	CPF	Produto
1	999999999	Caneta
2	999999999	Livro
3	888888888	Borracha

CPF	Nome	Endereço
999999999	Thiago	SQSW 302
888888888	Flávia	SQSW 304

Agora temos duas relações com ausência de dependência funcional parcial. Logo, nosso modelo acima está na segunda forma normal. Lembrando que existe uma cumulatividade entre as formas. Assim, para estar na segunda forma normal, a relação tem que estar na 1FN e cumprir os requisitos para eliminar as dependências parciais.

Uma definição mais rigorosa descreve que um esquema de relação R está na 2FN se todo atributo **não primário** A em R tem **dependência funcional total** de uma **chave candidata**. Podemos dizer também que **não existe dependência parcial**. De uma forma mais simples, a ideia aqui é que cada atributo não chave seja definido por todos os atributos pertencentes à chave primária ou a outra chave candidata da relação. Veja abaixo uma relação que não se adequa a segunda forma normal, sendo, por meio do processo de normalização, transformada em duas relações que estão de acordo com a definição apresentada.



N_pedido	Codigo_produto	Produto	Quant	Valor_unit	Subtotal
1005	1-934	Impressora laser	5	1.500,00	7.500,00
1006	1-956	Impressora desjet	3	350,00	1.050,00
1007	1-923	Impressora matricial	1	190,00	190,00
1008	1-908	Impressora mobile	6	980,00	5.880,00

N_pedido	Codigo_produto	Quant	Valor_unit	Subtotal
1005	1-934	5	1.500,00	7.500,00
1006	1-956	3	350,00	1.050,00
1007	1-923	1	190,00	190,00
1008	1-908	6	980,00	5.880,00

Codigo_produto	Produto
1-934	Impressora laser
1-956	Impressora desjet
1-923	Impressora matricial
1-908	Impressora mobile

Um fato interessante sobre a segunda forma normal é que para termos problema com uma relação precisamos que **uma chave candidata seja composta**, ou seja, possuir mais de um atributo, e que um desses atributos determine outro (atributo não primário) desta relação. Nas provas de concursos o que geralmente acontece é termos uma chave primária composta e um atributo pertencente a essa chave que determina um atributo não primário da tabela. Desta forma, é possível aceitar algumas das definições abaixo:

Segunda forma normal

- Um esquema de relação R está na 2FN se **cada atributo não primário A em R for total e funcionalmente dependente da chave primária** de R.
- Uma tabela está na segunda forma normal (2FN) se estiver na primeira forma normal (1FN), e seus atributos não chaves forem totalmente dependentes da chave primária.

TERCEIRA FORMA NORMAL (3FN)

A normalização feita a partir da regra definida pela terceira forma normal leva a relação para um estado específico. Neste caso, a relação tem que estar na segunda forma normal e ainda **todo atributo não primário da relação não ser transitivamente dependente de uma chave da relação**.

Uma relação está na Terceira Forma Normal (3FN) se ela estiver na 2FN e nenhum atributo não chave (não primário) é transitivamente dependente de uma chave candidata. Enfim, **na 3FN não se aceita dependência transitiva**. O Navathe descreve uma definição mais geral da terceira forma normal que diz basicamente o seguinte:

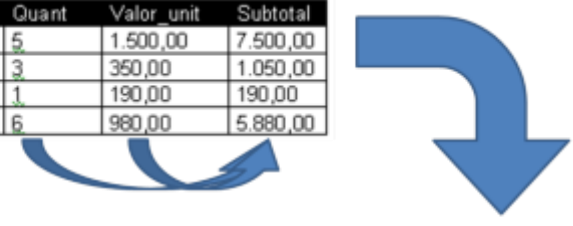
Um esquema de relação R está na terceira forma normal (3FN) sempre que uma dependência funcional não trivial $X \rightarrow A$ for determinada em R, qualquer

- X é superchave de R; ou
- A é atributo primário de R;

Segundo o próprio autor, violar a condição (a) significa que X não é um super conjunto de nenhuma chave de R; conseqüentemente, X pode ser não primário ou pode ser um dado subconjunto de uma chave de R. O autor fala também que a violação de (b) significa que A é um atributo não primário. Enfim, são condições para a terceira forma normal (1) ter

dependência funcional total para todas as chaves de R e (2) não ser transitivamente dependente de nenhuma chave de R.

N_pedido	Codigo_produto	Quant	Valor_unit	Subtotal
1005	1-934	5	1.500,00	7.500,00
1006	1-956	3	350,00	1.050,00
1007	1-923	1	190,00	190,00
1008	1-908	6	980,00	5.880,00



N_pedido	Codigo_produto	Quant	Valor_unit
1005	1-934	5	1.500,00
1006	1-956	3	350,00
1007	1-923	1	190,00
1008	1-908	6	980,00

Se você achou essa definição de dependência transitiva muito complexa e não conseguiu fixar nada, deixa eu tentar explicar de outra forma. Primeiro você precisa ter em mente que para existir a transitividade temos que ter algumas premissas. Um atributo chave (primário), por exemplo, CPF determina um outro atributo (não primário), por exemplo, telefoneResidencial; que por sua vez determina outro atributo (não primário), por exemplo, Endereco. Vejamos a tabela abaixo:

CPF	Nome	telResidencial	Endereco
001	Thiago	61 555-1255	SQSW 302 BL G
002	Flavia	61 555-1255	SQSW 302 BL G
003	Lucas	61 555-1555	SQSW 302 BL G
004	Vinicius	61 555-1555	SQSW 302 BL G
005	Ladjane	81 555-9299	Av. Portugal

Pense da seguinte forma: se você me passar um número de CPF e eu devolvo um telefone residencial. Da mesma forma, se você me der um número de telefone eu devolvo um endereço único. Observe que, alguns telefones aparecem mais de uma vez na coluna, contudo, eles determinam o mesmo endereço, ou, em outras palavras, eles estão associados ao mesmo endereço. É justamente essa replicação que desejamos evitar na terceira forma normal. Vamos então separar a tabela acima:

CPF	Nome	telResidencial
001	Thiago	61 555-1255
002	Flavia	61 555-1255
003	Lucas	61 555-1555
004	Vinicius	61 555-1555
005	Ladjane	81 555-9299

Veja que a redundância anterior desapareceu. Agora temos o telefone residência associado a apenas um endereço da tabela abaixo.

telResidencial	Endereco
----------------	----------



61 555-1255	SQSW 302 BL G
81 555-9299	Av. Portugal

Agora, voltando aos conceitos mais formais, você deve lembrar que um atributo primário faz parte de alguma chave candidata da sua relação. E para que exista necessidade de normalizar utilizando a terceira forma normal temos que ter 3 atributos, vamos supor A, B e C, de forma que A seja **primário** e B e C **não primários**. Além disso, temos que ter uma dependência transitiva onde $A \rightarrow B$ e $B \rightarrow C$.

Antes de apresentarmos as outras formas normais vamos fazer uma questão sobre os conceitos vistos até aqui:



Ano: 2017 Órgão: Alerj Cargo: Analista de Tecnologia da Informação Q. 48

Em banco de dados, a finalidade do processo de normalização é evitar redundâncias e, portanto, evitar certas anomalias de atualização de dados. Considere as dependências funcionais entre os atributos das seguintes entidades:

PACIENTE(ID_PACIENTE determina NOME_PACIENTE);

MEDICO(ID_MEDICO determina CRM_MEDICO, NOME_MEDICO);

CONSULTA(ID_PACIENTE, ID_MEDICO determinam DATA_ATEND, HORA_ATEND);

Sabendo-se que o atributo sublinhado é a chave primária, a alternativa que apresenta as entidades e seus atributos na Terceira Forma Normal (3FN) é:

(A) PACIENTE (ID_PACIENTE, NOME_PACIENTE, ID_MEDICO, DATA_ATEND, HORA_ATEND)

MEDICO (ID_MEDICO, CRM_MEDICO, NOME_MEDICO)

CONSULTA (CRM_MEDICO, DATA_ATEND, HORA_ATEND)

(B) PACIENTE (ID_PACIENTE, NOME_PACIENTE)

MEDICO (ID_MEDICO, CRM_MEDICO, NOME_MEDICO)

CONSULTA (ID_PACIENTE, NOME_MEDICO, DATA_ATEND, HORA_ATEND)

(C) PACIENTE (ID_PACIENTE, NOME_PACIENTE, ID_MEDICO)

MEDICO (ID_MEDICO, CRM_MEDICO, NOME_MEDICO)

CONSULTA (ID_PACIENTE, DATA_ATEND, HORA_ATEND)

(D) PACIENTE (ID_PACIENTE, NOME_PACIENTE)

MEDICO (ID_MEDICO, CRM_MEDICO, NOME_MEDICO)

CONSULTA (ID_PACIENTE, ID_MEDICO, DATA_ATEND, HORA_ATEND)

(E) PACIENTE (ID_PACIENTE, NOME_PACIENTE)

MEDICO (ID_MEDICO, CRM_MEDICO, NOME_MEDICO)

CONSULTA (ID_PACIENTE, CRM_MEDICO, NOME_MEDICO, DATA_ATEND, HORA_ATEND)



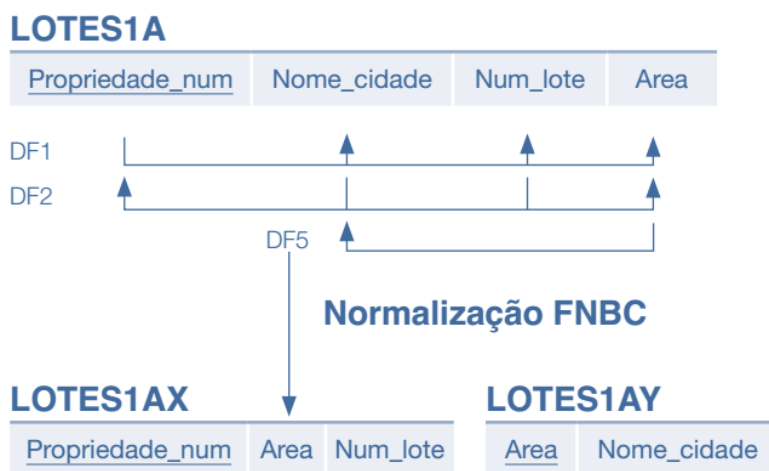
Comentário: Para chegarmos até a terceira forma normal temos que cumprir alguns requisitos. As relações do modelo não podem ter atributos compostos ou multivalorados (1FN), não pode existir em cada uma das relações dependência parcial (2FN) nem dependência transitiva (3FN). Ao ajustar o modelo podemos encontrar nossa resposta na alternativa D.

Gabarito: D

FORMA NORMAL DE BOYCE-CODD (FNBC)

Uma coisa interessante é que a forma normal de Boyce-Codd foi proposta como **uma forma mais simples que 3FN, porém mais rígida**. Devido ao fato de a 3FN não tratar satisfatoriamente casos onde uma relação tem mais de uma chave candidata, e quando estas chaves são compostas e possuem atributos em comum.

Se uma relação está na FNBC, também está na 3FN. Sua definição diz o seguinte: **uma relação está na FNBC se todo determinante é chave candidata**. Abaixo segue uma figura que demonstra uma normalização de uma relação para a FNBC.



Veja na figura que temos duas chaves candidatas (Propriedade_num) e (Nome_cidade, Num_lote). Temos ainda uma dependência funcional onde Area determina o Nome_cidade. Neste caso, temos que fazer a separação proposta na figura acima. Veja que Area → Nome_cidade não fere a terceira forma normal pois Nome_cidade é um atributo primário. Vamos agora continuar nosso estudo fazendo mais uma questão sobre o assunto.



Analista de Políticas Públicas e Gestão Governamental - CGM Niterói - 2018

A identificação das dependências funcionais constitui um importante passo para a normalização de tabelas de bancos de dados. Considere uma tabela T, com atributos A, B e C, onde A foi definido como primary key, e C como unique.



Assinale a opção que indica o mínimo conjunto de dependências funcionais que devem existir, além das dependências triviais e das que podem ser derivadas, para que essa tabela esteja normalizada até a forma normal Boyce-Codd.

- a) $A \rightarrow B$; $A \rightarrow C$; $B \rightarrow C$.
- b) $A \rightarrow B$; $B \rightarrow C$.
- c) $C \rightarrow A$; $C \rightarrow B$.
- d) $A \rightarrow B$; $A \rightarrow C$.
- e) $A \rightarrow B$; $A \rightarrow C$; $B \rightarrow A$.

Comentário: Essa questão pode ter sua resolução dividida em duas partes. Sabemos que temos três atributos: A, B e C. O atributo "**A**" foi definido como **chave primária**. Logo, pela explicação que vimos até aqui temos que **$A \rightarrow B$** e **$A \rightarrow C$** . A chave primária não pode ter valores nulos

A outra informação que temos é que o atributo C possui a propriedade de **unicidade (UNIQUE)**. Neste caso temos que conhecer um pouco de SQL para facilitar a resolução da questão. O fato de um atributo ser definido como unique não faz com que ele seja um determinante, pois **o valor nulo continua fazendo parte do seu domínio**. Perceba que, se o valor nulo pode aparecer em uma linha associado ao atributo C, C não pode ser definido como um determinante. Assim, C é um atributo não chave ou não primário.

Agora vamos relembrar da definição de FNBC: "todo determinante é chave candidata." O atributo A já é uma chave candidata. Nossa preocupação agora é com B. B precisar ser um determinante e uma chave candidata para não termos problema com a FNBC nem com a 3FN. Logo, temos que garantir que **$B \rightarrow A$** . Com essa dependência, B passa a ser chave candidata pois determina a chave primária. Assim, nosso conjunto de dependência resolve nosso problema.

Mas professor, e a alternativa A, por que ela não é a nossa resposta visto que $B \rightarrow C$? Porque, neste caso, temos uma dependência transitiva, visto que B e C são não chave. Logo, na letra A podemos visualizar uma dependência transitiva $A \rightarrow B$ e $B \rightarrow C$, o que fere a terceira forma normal. Já na alternativa E esse problema está resolvido pois tanto A quanto B são chaves candidatas, então, mesmo que C não seja um atributo não chave ele não vai gerar problemas com a FNBC.

Logo, temos a nossa resposta na alternativa E.

Gabarito: E



QUESTÕES COMENTADAS CESPE

QUESTÕES COMENTADAS

Aproveitamos este espaço para complementar seu entendimento sobre o assunto, sempre colocando pinceladas extras de conteúdo. Na lista abaixo resolvemos incluir apenas questões do CESPE. Essas questões abordam os assuntos vistos nesta nossa aula: **modelo relacional**, **álgebra relacional** e **formas normais**. A minha sugestão é que, antes de tentar fazer as questões você assista aos vídeos sobre o assunto associados a essa aula.



1. Ano: 2018 Banca: CESPE Assunto: Informática para Polícia Federal Cargo: Escrivão Conteúdo Banco de dados Relacional

CPF
NOME
DATA DE NASCIMENTO
NOME DO PAI
NOME DA MAE
TELEFONE
CEP
NUMERO

As informações anteriormente apresentadas correspondem aos campos de uma tabela de um banco de dados, a qual é acessada por mais de um sistema de informação e também por outras tabelas. Esses dados são utilizados para simples cadastros, desde a consulta até sua alteração, e também para prevenção à fraude, por meio de verificação dos dados da tabela e de outros dados em diferentes bases de dados ou outros meios de informação. Considerando essas informações, julgue os itens que se seguem.

93 Se um sistema de informação correlaciona os dados da tabela em questão com outros dados não estruturados, então, nesse caso, ocorre um processo de mineração de dados.

94 A referida tabela faz parte de um banco de dados relacional

95 O campo CPF é utilizado como chave primária e chave estrangeira.

96 Os dados armazenados na referida tabela são considerados não estruturados.

Comentário:



93. Essa questão foi dada pelo CESPE como correta. Eu discordo. Vamos lá ... supondo que exista uma tabela (modelo relacional) com dados e que os dados são correlacionados com outros dados não estruturados, que garantia eu tenho que os processos de mineração de dados foi utilizado de fato utilizados? ... NENHUMA!!!

O processo de mineração de dados, como descrito pelo CRISP-DM (fonte de referência padrão para o assunto) exige um conjunto de ações específicas. Não existe nenhum elemento no texto, além da detecção de fraudes que possa nos remeter a mineração de dados. A mineração de dados exige que os dados sejam transformados e organizados com um mínimo de estrutura para que possam ser analisados com sucesso.

Para mim, a correlação de dados estruturados e não estruturados é associada ao processo de Big Data, a mineração de dados como sugere a alternativa, pode aparecer como parte desse processo. Logo, na minha opinião temos uma alternativa **errada, mas o CESPE julgou CORRETA**.

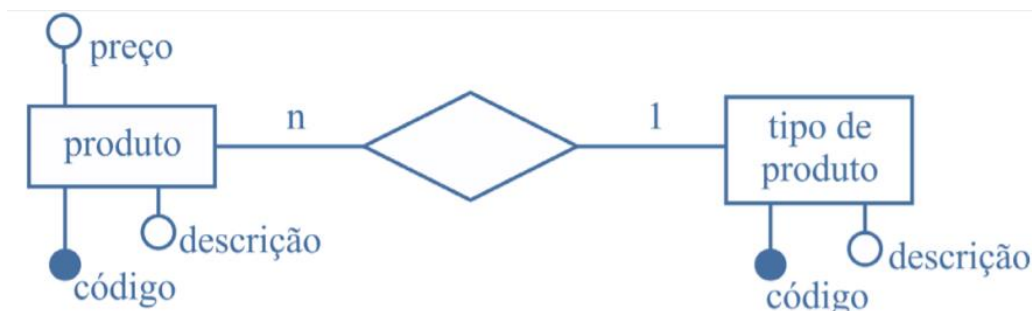
94. Para que a tabela seja considerada uma relação ela tem que possuir atributos atômicos ou estar na primeira forma normal. Como não temos atributos compostos ou multivalorados podemos considerar a normalização a 1FN. Sendo assim, podemos marcar a assertiva como **correta**.

95. O campo CPF pela descrição organização dos atributos pode ser usado como chave primária, contudo não faz sentido falar de chave estrangeira quando temos apenas uma relação. Logo temos mais uma alternativa **incorreta**.

96. Os dados armazenados no modelo relacional são dados estruturados. Logo, a alternativa está **incorreta**.

Gabarito: 93.C(com ressalvas) 94.C 95.E 96.E

2. Ano: 2018 Banca: CESPE Assunto: Informática para Polícia Federal Cargo: Agente Conteúdo Banco de dados



Considerando o modelo entidade-relacionamento (ER) precedente, julgue os seguintes itens, relativos a banco de dados.

82 Considerando-se apenas o diagrama apresentado, infere-se que, na aplicação das regras para a transformação do modelo ER em um modelo relacional, é necessário realizar a fusão das tabelas referentes às entidades envolvidas no relacionamento.



Comentário: Vamos comentar cada uma das afirmações acima.

82. Quando vamos passar do modelo ER para o relacional um relacionamento 1:N, precisamos pegar a chave primária do lado 1 e colocar como chave estrangeira do lado N. No exemplo, temos que usar a chave primária (código) associada ao tipo de produto e inserir um novo campo na relação produto que será uma chave estrangeira e aponta para o código do tipo de produto (na tabela tipo de produto). A fusão de tabelas geralmente é usada quando temos um relacionamento obrigatório 1:1. Sendo assim, temos uma alternativa **incorreta**.

Gabarito: 82. E

3. Ano: 2018 Banca: CESPE Órgão: EBSE RH Prova: Analista de Tecnologia da Informação

Com relação a banco de dados, julgue o item seguinte.

Em normalização, a primeira forma normal é caracterizada por uma tabela com a existência obrigatória de uma chave primária e uma chave estrangeira.

Comentário: O gabarito da questão 1 é errado. A primeira forma normal (1FN) é a primeira etapa do processo de normalização que visa a redução da redundância e anomalias de atualização dos modelos de dados.

A 1FN vai remover do modelo os atributos compostos e multivalorados. Assim, todos os atributos das tabelas passam a ser atômicos.

A chave primária vai orientar o banco de dados quanto a organização dos dados em uma tabela. Ela tem a propriedade de identificar uma única linha da tabela. Em outras palavras, caso você me informe a chave eu consigo te devolver todos os demais valores dos atributos de uma tabela.

Já a chave estrangeira serve para fazer uma ligação entre duas tabelas. Se uma tabela tem um atributo que é chave estrangeira, significa que domínio (valores possíveis) desse atributo estão presentes uma coluna de outra tabela.

Gabarito: E

4. Ano: 2018 Banca: CESPE Órgão: TCM-BA Cargo: Auditor de Contas Questão: 09

Considerando os conceitos de banco de dados relacionais, assinale a opção correta a respeito das propriedades de uma tupla.

A A tupla tem o mesmo significado e as mesmas propriedades de uma tabela.

B Os componentes de uma tupla são ordenados da esquerda para a direita.

C Cada tupla contém exatamente um valor para cada um de seus atributos.

D Um subconjunto de uma tupla não é considerado uma tupla.

E Uma tupla nunca é vazia, seu grau pode variar de 1 até n.

Comentário: Vimos informações que as tuplas são linhas de uma tabela. Elas, quando consideramos a referência teórica e matemática do modelo relacional não



são ordenadas e não se repetem. Vamos agora analisar cada uma das alternativas acima.

A) A tupla é a linha da tabela. Representa uma instância ou um valor armazenado. Pense na tabela aluno, cada aluno armazenado é uma tupla da tabela. Contudo, uma tabela pode ter atributos que não são específicos da tupla como a quantidade máxima de registros. Logo, não podemos definir o todo pela parte, nem podemos dizer que a forma do bolo é um bolo. Sendo assim, a alternativa está incorreta.

B) Os as tuplas não são ordenadas de cima para baixo, nem seus atributos ou componentes são ordenados da esquerda para a direita.

C) Pela definição de modelo relacional do Codd os atributos de uma relação devem ter valores atômicos. Logo, cada coluna deve ter um valor dentro do seu respectivo domínio. Este pode ou não aceitar valores nulos. Sendo assim, essa afirmação está certa.

D) Um subconjunto de uma tupla é outra tupla, inclusive se lembrarmos da propriedade de fechamento das operações de álgebra relacional, podemos perceber que o resultado de uma projeção vai reduzir as tuplas de uma tabela aos atributos definidos na operação.

E) Uma tupla de uma tabela pode ser vazia (confesso que descobri isso com essa questão), ela é conhecida como *empty tuple* ou *nullary tuple*. Para preencher valores desconhecidos usamos o valor nulo que uma notação para vazio. Além disso ela representa um conjunto de atributos, a quantidade de atributos de uma tupla é denominada grau e pode variar de 0 até n. Logo, temos mais uma alternativa incorreta.

Gabarito: C

5. Ano: 2015 Banca: CESPE Órgão: STJ Prova: Analista Judiciário - Análise de Sistemas de Informação

Acerca de modelagem relacional e pontos de função, julgue o item a seguir.

O modelo relacional consiste em uma coleção ilimitada de tipos escalares e de um operador de atribuição relacional que atribui valores às variáveis de relações que integram os componentes desse modelo.

Comentário: OK! O texto em questão parece Grego. Mas não é! Você já precisa apenas de um tradutor para resolver a questão. Quando falamos em tipos **escalares**, estamos nos referindo a atributos que tem valores **atômicos e indivisíveis**. Numa classificação normal um escalar se contrapõe a uma matriz. Em uma matriz temos vários valores associados a uma instância.

Agora já temos uma coleção de tipos escalares que estabelecem os domínios dos atributos que formam uma tabela ou uma relação. Certo!? Assim, construímos uma tabela! Vamos precisar, então, inserir novos valores nesta tabela. Neste contexto, precisamos de um operador que permita a inserção de novas linhas. Esse operador de atribuição vai permitir que essa associação seja feita.



Vamos lembrar da nossa videoaula. Nela apresentamos um conjunto de componentes definidos pelo Date para banco de dados relacionais. Segundo ele o modelo relacional consiste em cinco componentes:

- 1) Uma **coleção ilimitada de tipos escalares**, incluindo em particular o tipo booleano ou valor verdade. (TIPOS DE DADOS)
- 2) Um gerador de tipo de relação e uma interpretação pretendida para esses tipos de relações. (A EXISTÊNCIA DAS TABELAS)
- 3) Recursos para definição de RelVars desses tipos de relações gerados. (DDL – DEFINIÇÃO – UMA LINGUAGEM PARA CONSTRUIR AS TABELAS)
- 4) Um operador de atribuição relacional para atribuição de valores de relações a essas RelVars. (DML – MANIPULÇÃO/INSERTÃO DE VALORES NAS TABELAS)
- 5) Uma **coleção ilimitada de operadores relacionais** genéricos para derivar valores de relações a partir de outros valores de relações. (OPERAÇÕES)

Gabarito: C.

6. Banca: CESPE Ano: 2015 Órgão: TRE-MT Prova: Analista Judiciário - Análise de Sistemas

No modelo relacional formal,

- a) os elementos de uma relação respeitam uma ordem matemática entre eles.
- b) cada coluna em uma relação é uma tupla.
- c) cada cabeçalho em uma relação é uma chave.
- d) domínio é um conjunto de valores em que cada valor é indivisível.
- e) uma coleção de dados é considerada como um arquivo plano.

Comentário: Vamos retomar os conceitos para que você consiga fixar melhor cada um deles. Os elementos de uma relação são as linhas (tuplas) e colunas (atributos), não existe uma ordem lógica sobre as linhas, elas podem ser armazenadas em qualquer ordem na tabela.

Já na alternativa B o erro está em definir uma coluna como uma tupla. Sabemos que uma tupla é a linha da tabela! Lembre-se [do funk da tabela](#)! Cada cabeçalho descreve o nome do atributo referente a coluna. Não existe obrigatoriedade de um atributo ser chave. Sendo assim, a alternativa C também está incorreta.

A alternativa D é a nossa resposta! Veja que o domínio é um escalar, logo indivisível. Cada tipo de dados que estabelece um domínio vai restringir o conjunto de valores de uma coluna. Por exemplo, você pode permitir que apenas caracteres sejam incluídos na coluna Nome.

Um arquivo plano é um arquivo tipo .txt. Semelhante aquele que você cria no Notepad do Windows. A ideia de definir uma coleção de dados como um arquivo TXT não me parece saudável. Sendo assim, a alternativa E está incorreta.



Gabarito: D.

7. BANCA: CESPE ANO: 2015 ÓRGÃO: TRE-GO PROVA: TÉCNICO DO JUDICIÁRIO - PROGRAMAÇÃO DE SISTEMAS

Julgue os seguintes itens, a respeito da modelagem de dados.

[65] Considere a seguinte situação hipotética. Em um banco de dados referente a um curso, um aluno pode estar em mais de um curso ao mesmo tempo. Além disso, na tabela de cursos realizados por aluno, estão presentes as chaves estrangeiras aluno e curso. Nessa situação, tanto o código do curso como o código do aluno são chaves primárias nas tabelas curso e aluno, respectivamente.

[66] Ao se excluir uma tupla de um banco de dados, pode-se violar a integridade referencial desse banco por uma chave primária.

[67] Um conjunto de entidades que não possuem atributos suficientes para formar uma chave primária é definido como um conjunto de entidades fortes.

[68] Uma chave primária identifica um único valor de uma tupla no banco de dados e não possui mais de um atributo na tabela.

Comentário. Vamos comentar cada um dos itens acima.

[65] Analisando a alternativa 65 podemos observar que trata de uma narrativa consistente e, portanto, **correta**.

[66] Na questão 66 temos um **erro** ao disser que a exclusão de uma linha pode violar a integridade referencial por meio da chave primária, o certo seria dizer que pode existir uma violação de integridade por meio da chave estrangeira. Em outras palavras podemos pensar da seguinte forma: eu só posso excluir uma linha da tabela A se não existir nenhuma referência a ela em outra tabela X. Essa referência é feita por meio de uma chave estrangeira, presente na outra tabela X.

[67] Na questão 67 o **erro** é atribuir a entidade forte o conceito de entidade fraca.

[68] A alternativa 68 trata da chave primária de uma tabela. Essa pode ser definida sobre um ou vários atributos. Quando definida sobre mais de um atributo é necessário que a informação contida no conjunto de atributos da chave seja única para cada linha da tabela. Sendo assim podemos assinalar a alternativa como **incorreta**.

Gabarito: C E E E.

8. Ano: 2015 Banca: CESPE Órgão: STJ Prova: Analista Judiciário - Análise de Sistemas de Informação

Acerca de modelagem relacional, julgue o item a seguir.

O modelo relacional de dados consiste em um banco de dados percebido por seus usuários como uma coleção de variáveis de relações que trata das questões lógicas e físicas da estrutura, da integridade e da manipulação de dados.



Comentário: Essa questão trata de um ponto que eu gostaria muito que você guardasse para prova: modelo relacional é um modelo **lógico**! Não é físico como sugere a questão. Modelos físicos fazem parte da implementação dos sistemas de banco de dados. São definidos por programadores que constroem os SGBD, algo complexo até para mim que trabalho com banco de dado há alguns anos. Sendo assim, a questão está equivocada ao tentar associar o modelo relacional ao nível físico.

Agora vamos tentar esclarecer outros pontos interessantes do texto. O modelo relacional define o modelo como uma coleção de tabelas ou variáveis de relações. Veja que essas estruturas, as tabelas, são de fato variáveis pois mudam ao longo do tempo. Pense que os preços dos produtos em um supermercado variam, os funcionários contratados em uma empresa também se alteram ao longo do tempo. Logo, as tabelas que armazenam essas informações são modificadas para manter a consistência com os eventos de negócio, como a contratação de um novo funcionário.

Gabarito: E

9. Ano: 2010 Banca: CESPE Órgão: MPU Prova: Analista de Informática - Banco de Dados

Acerca de administração de banco de dados relacionais, julgue os itens que se seguem.

O termo integridade é utilizado em sistema de banco de dados com o significado de precisão, correção ou validade. Nesse contexto, a integridade tem como função assegurar que os dados no banco de dados sejam precisos e preservados contra atualizações válidas.

Comentário: Essa questão apresenta uma pegadinha. Esse é um tipo de questão para as pessoas que sofrem do mal da leitura dinâmica rústica errarem. Muita calma ao ler as questões do seu concurso. A afirmação: “preservado contra atualizações válidas” distorce uma das funções do SGBD. O SGBD deve ser utilizado para garantir a consistência do banco de dados. Preservando contra anomalias de atualização ou atualizações inválidas. Logo, a alternativa está incorreta.

Gabarito: E.

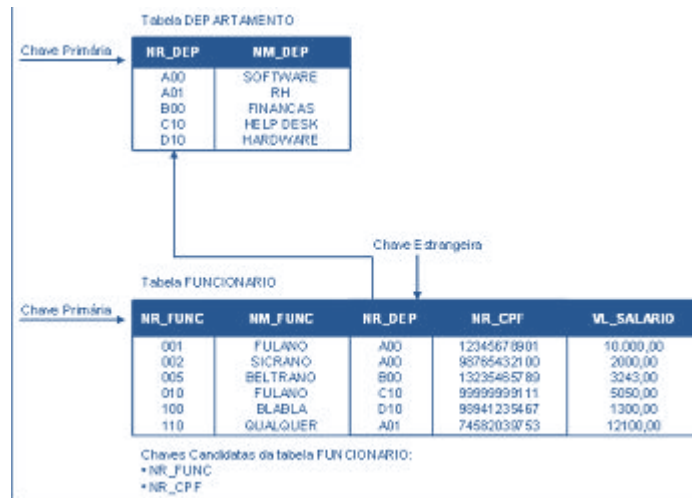
10. Ano: 2008 Banca: CESPE Órgão: STF Prova: Analista Judiciário - Tecnologia da Informação

O armazenamento e a recuperação de grandes quantidades de dados é um trabalho importante e muito explorado em um sistema gerenciador de banco de dados (SGBD). Com relação aos conceitos que envolvem esse sistema, julgue os itens que se seguem.

Integridade referencial pode ser definida como uma condição imposta a um conjunto de atributos de uma relação para que valores que apareçam nesse conjunto também apareçam em um certo conjunto de atributos de uma outra relação.



Comentário: Essa é uma questão um pouco antiga, tem quase 10 anos. Contudo, apresenta o conceito de integridade referencial. Como eu falei, a integridade referencial (IR) permite que o relacionamento entre entidades do modelo E-R seja expresso no modelo relacional. A ligação entre as tabelas é feita por meio destas IRs. Vejam mais um exemplo desta situação na figura abaixo:



Gabarito: C.

11. Ano: 2014 Banca: CESPE Órgão: SUFRAMA Prova: Analista Técnico - Tecnologia da Informação

Com relação aos sistemas gerenciadores de banco de dados (SGBD), julgue os itens a seguir.

A integridade semântica de um SGBD garante que os dados estejam sempre corretos em relação ao domínio de aplicação.

Comentário: A integridade semântica tem relação direta com as regras de negócio, ou seja, são restrições impostas pelo funcionamento da organização. Um gerente não pode ganhar mais do que um diretor, por exemplo. Essas restrições garantem que os dados estejam corretos em relação ao domínio da aplicação. Sendo assim, a alternativa está correta.

Gabarito: C.

12. BANCA: CESPE ANO: 2015 ÓRGÃO: TJDF PROVA: PROGRAMAÇÃO DE SISTEMAS

Julgue os itens seguintes a respeito de banco de dados.

[61] Em uma tabela de um banco de dados relacional, se uma restrição de chave primária for definida como composta de mais de uma coluna, os seus valores poderão ser duplicados em uma coluna; no entanto, cada combinação de valores de todas as colunas na definição da restrição de chave primária deve ser exclusiva.



Comentário: Questão interessante, pois nos permite fazer um rápido comentário sobre chaves. Uma **chave** identifica unicamente uma linha de uma relação. Toda relação pode ter vários conjuntos de atributos que podem ser escolhidos como **chave primária**. Cada uma dessas opções que se caracterizam por ser uma **superchave mínima**, ou seja, não é possível retirar nenhum atributo sem que o conjunto perca a propriedade de ser chave da relação, é denominada **chave candidata**. A chave escolhida para ser a chave da relação é denominada chave primária. Ela pode ser **composta** por **um ou mais atributos**.

A questão pede para analisarmos uma característica de chaves compostas por mais de um atributo. Percebam que a unicidade dos valores deve considerar o conjunto dos atributos e não um atributo individualmente. Sendo assim a alternativa encontra-se **correta**.

Gabarito: C.

13. Ano: 2018 Banca: CESPE Órgão: STJ Cargo: Técnico Judiciário – Suporte Técnico

Acerca de banco de dados, julgue os itens que se seguem.

73 Relacionamentos do tipo um-para-um podem ser representados em até três tabelas, de acordo com a obrigatoriedade do relacionamento.

77 Na criação de uma tabela para os clientes de uma organização, os atributos de CPF e CNPJ, para pessoas físicas e jurídicas, respectivamente, são a escolha mais indicada para representar a chave primária (PK) da tabela.

Comentário: Uma das soluções para os relacionamentos 1:1, em especial quando não existe obrigatoriedade, é criar uma tabela de ligação para armazenar as instâncias do relacionamento. Sendo assim, podemos marcar nossa resposta para a afirmação 73 como correta.

Já a afirmação 77, seria mais interessante criar uma chave artificial para representar o identificador do cliente. Lembre-se que CPF e CNPJ tem tamanho distintos e diversas peculiaridades para validação do número. Assim a afirmação está incorreta.

Gabarito: C E.

14. Ano: 2018 Banca: CESPE Órgão: ABIN Cargo: Área 08 Questão: 142

A respeito de sistemas gerenciadores de banco de dados, julgue os próximos itens.

142 Chave primária é o conjunto de um ou mais atributos para identificar uma tupla de uma entidade.

Comentário: O atributo (coluna) ou combinação de atributos que identifica de forma exclusiva cada tupla (linha) de uma relação (tabela) é chamada de **chave primária**. Uma relação pode ter apenas uma chave primária, mas pode ser a combinação de mais de um atributo. A chave primária não pode conter valores **duplicados ou nulos**. Se a chave primária for uma combinação de mais de um atributo, **a combinação de**



todos os valores de atributo deve ser exclusiva e nenhum dos atributos pode conter valor nulo. Nulo significa valor desconhecido. Não significa zero ou espaço, ou em branco. Desta forma a alternativa está correta.

Gabarito: C.

15. Ano: 2018 Banca: CESPE Órgão: STM Cargo: Programação de Sistemas
Questão: 61 a 65

Acerca dos conceitos de normalização de dados e dos modelos de dados, julgue os itens subsequentes.

61 Uma tabela estará na segunda forma normal (2FN) quando, além de estar na terceira forma normal (3FN), ela contiver dependências funcionais parciais.

62 A passagem à terceira forma normal (3FN) tem como objetivo principal gerar o modelo lógico de dados; por isso, ela não visa eliminar redundância de dados, como ocorre com as demais formas normais.

65 A transformação do esquema de tabela não normalizada em um esquema relacional na primeira forma normal (1FN) consiste da eliminação das tabelas aninhadas.

Comentário: Vamos comentar cada uma das alternativas acima.

61. Primeiramente a hierarquia entre as formas normais é crescente, para estar na segunda é preciso estar na primeira, para estar na terceira é necessário que esteja na segunda e assim por diante. Veja que a afirmação da alternativa vai no sentido oposto. Logo, temos uma alternativa incorreta.

62. A passagem para a terceira forma normal tem como objetivo eliminar as dependências funcionais transitivas. O que está escrito na assertiva não chegar nem perto da definição. Sendo assim, a alternativa está errada.

65. Essa questão do aninhamento de tabela associada a primeira forma normal nem sempre é muito clara. Contudo, podemos perceber que, quando temos atributos compostos e multivalorados, há perda de atomicidade dos atributos. E a composição de diversos atributos atômicos pode ser vista como outra tabela. Logo, acabamos por ter uma tabela dentro da outra.

Gabarito: E E C.

16. BANCA: CESPE ANO: 2015 ÓRGÃO: CGE-PI PROVA: AUDITOR GOVERNAMENTAL - TECNOLOGIA DA INFORMAÇÃO

A respeito de banco de dados, julgue os itens subsequentes.

Para normalizar, conforme primeira forma, uma tabela em um banco de dados, é preciso criar chaves estrangeiras que representem a ligação entre elas.

Comentário: A primeira forma normal deve eliminar atributos compostos e multivalorados. Vejam que no caso dos atributos compostos não é necessário criar



outra relação, basta fracionar as partes em atributos atômicos. Quando estamos removendo atributos multivalorados podemos utilizar outra tabela para listar os diversos valores e associá-los por meio da chave. Ou seja, nem sempre é necessário usar chaves estrangeiras quando normalizamos para a 1FN.

Gabarito: E.

17. BANCA: CESPE ANO: 2013 ÓRGÃO: CRPM PROVA: ANALISTA EM GEOCIÊNCIAS - SISTEMAS DE INFORMAÇÃO

No que concerne a mapeamento de dados lógico e físico e a elaboração e implantação de projeto de banco de dados, julgue os seguintes itens.

No processo de implantação de um projeto de banco de dados, devem ser utilizadas as operações de álgebra relacional de dados para estabelecer as restrições de cardinalidade e relacionamento entre o conjunto de entidades.

Comentário: As operações da álgebra relacional fornecem subsídio matemático para execução de operações no modelo relacional. Elas são utilizadas pelos desenvolvedores de SGBDs para estruturar as operações sobre as tabelas em bancos de dados relacionais.

A cardinalidade entre os relacionamentos e os relacionamentos em si são implementados por meio das restrições de integridade e da organização ou estruturação dos dados nas diversas tabelas do modelo. Um relacionamento N para N precisa de uma tabela de ligação que é formada pelas chaves de cada uma das tabelas que participam do relacionamento.

Gabarito: E.

18. BANCA: CESPE ANO: 2013 ÓRGÃO: ANTT PROVA: ANALISTA ADMINISTRATIVO - DESENVOLVIMENTO DE SISTEMAS

Julgue os itens subsequentes, relativos a banco de dados.

A linguagem padrão de consulta SQL (structured query language) utiliza uma combinação de construtores em álgebra e cálculo relacional.

Comentário: O cálculo relacional é considerado a base para linguagem SQL, e a álgebra relacional é usada nos detalhes internos de muitas implementações de banco de dados para processamento e otimização de consulta. Alternativa correta!

Gabarito: C.



19. BANCA: CESPE ANO: 2013 ÓRGÃO: TCE-ES PROVA: ANALISTA ADMINISTRATIVO - INFORMÁTICA



O conjunto de operações cujo resultado seja uma nova relação e que envolve seleção, projeção, união e produto cartesiano é denominado

A mapeamento de cardinalidades.

B álgebra relacional.

C generalização.

D chave primária.

E herança.

Comentário: Essa questão você deve fazer rapidamente e ganhar tempo na hora da prova para outras questões mais complexas. Vejam que ela basicamente quer saber se você conhece as operações da álgebra relacional.

Gabarito B.

20. CESPE - Auditor de Controle Externo (TCE-PA)/Informática/Analista de Sistema/2016

Julgue o item que se segue, relativo a modelagem de dados.

Dois diagramas de entidade de relacionamento são equivalentes se possuem entidades e relacionamentos que geram o mesmo esquema de banco de dados.

Comentário: Um diagrama E-R descreve o modelo de dados de um sistema com um grande nível de abstração, independentemente do SGBD a ser escolhido. Um esquema de bancos de dados é a descrição da estrutura em uma linguagem formal suportada pelo SGBD e refere-se à organização de dados (dividido em tabelas no caso de BDs relacionais). Logo, mesmo que diagramas E-R sejam diferentes, eles são considerados equivalentes caso gerem o mesmo esquema de banco de dados.

Gabarito: C



EXERCÍCIOS



1. Ano: 2018 Banca: CESPE Assunto: Informática para Polícia Federal Cargo: Escrivão Conteúdo Banco de dados Relacional

CPF
NOME
DATA DE NASCIMENTO
NOME DO PAI
NOME DA MAE
TELEFONE
CEP
NUMERO

As informações anteriormente apresentadas correspondem aos campos de uma tabela de um banco de dados, a qual é acessada por mais de um sistema de informação e também por outras tabelas. Esses dados são utilizados para simples cadastros, desde a consulta até sua alteração, e também para prevenção à fraude, por meio de verificação dos dados da tabela e de outros dados em diferentes bases de dados ou outros meios de informação. Considerando essas informações, julgue os itens que se seguem.

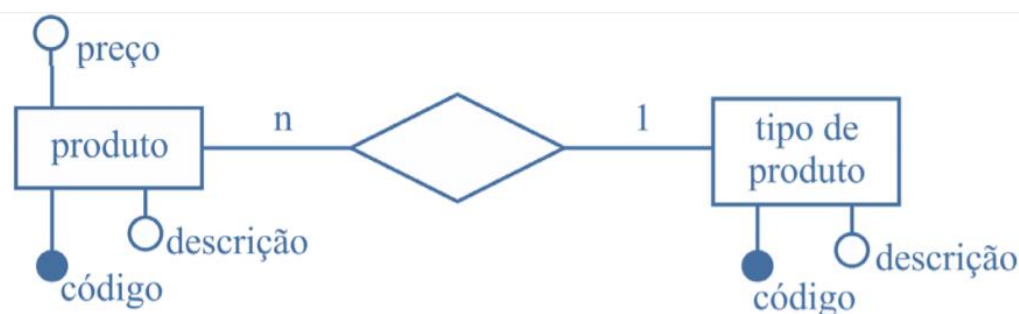
93 Se um sistema de informação correlaciona os dados da tabela em questão com outros dados não estruturados, então, nesse caso, ocorre um processo de mineração de dados.

94 A referida tabela faz parte de um banco de dados relacional

95 O campo CPF é utilizado como chave primária e chave estrangeira.

96 Os dados armazenados na referida tabela são considerados não estruturados.

2. Ano: 2018 Banca: CESPE Assunto: Informática para Polícia Federal Cargo: Agente Conteúdo Banco de dados



Considerando o modelo entidade-relacionamento (ER) precedente, julgue os seguintes itens, relativos a banco de dados.

82 Considerando-se apenas o diagrama apresentado, infere-se que, na aplicação das regras para a transformação do modelo ER em um modelo relacional, é necessário realizar a fusão das tabelas referentes às entidades envolvidas no relacionamento.

3. Ano: 2018 Banca: CESPE Órgão: EBSEH Prova: Analista de Tecnologia da Informação

Com relação a banco de dados, julgue o item seguinte.

Em normalização, a primeira forma normal é caracterizada por uma tabela com a existência obrigatória de uma chave primária e uma chave estrangeira.

4. Ano: 2018 Banca: CESPE Órgão: TCM-BA Cargo: Auditor de Contas Questão: 09

Considerando os conceitos de banco de dados relacionais, assinale a opção correta a respeito das propriedades de uma tupla.

- A A tupla tem o mesmo significado e as mesmas propriedades de uma tabela.
- B Os componentes de uma tupla são ordenados da esquerda para a direita.
- C Cada tupla contém exatamente um valor para cada um de seus atributos.
- D Um subconjunto de uma tupla não é considerado uma tupla.
- E Uma tupla nunca é vazia, seu grau pode variar de 1 até n.

5. Ano: 2015 Banca: CESPE Órgão: STJ Prova: Analista Judiciário - Análise de Sistemas de Informação

Acerca de modelagem relacional e pontos de função, julgue o item a seguir.

O modelo relacional consiste em uma coleção ilimitada de tipos escalares e de um operador de atribuição relacional que atribui valores às variáveis de relações que integram os componentes desse modelo.

6. Banca: CESPE Ano: 2015 Órgão: TRE-MT Prova: Analista Judiciário - Análise de Sistemas

No modelo relacional formal,

- a) os elementos de uma relação respeitam uma ordem matemática entre eles.
- b) cada coluna em uma relação é uma tupla.
- c) cada cabeçalho em uma relação é uma chave.
- d) domínio é um conjunto de valores em que cada valor é indivisível.
- e) uma coleção de dados é considerada como um arquivo plano.

7. Ano: 2015 Banca: CESPE Órgão: STJ Prova: Analista Judiciário - Análise de Sistemas de Informação

Acerca de modelagem relacional, julgue o item a seguir.



O modelo relacional de dados consiste em um banco de dados percebido por seus usuários como uma coleção de variáveis de relações que trata das questões lógicas e físicas da estrutura, da integridade e da manipulação de dados.

8. Ano: 2010 Banca: CESPE Órgão: MPU Prova: Analista de Informática - Banco de Dados

Acerca de administração de banco de dados relacionais, julgue os itens que se seguem.

O termo integridade é utilizado em sistema de banco de dados com o significado de precisão, correção ou validade. Nesse contexto, a integridade tem como função assegurar que os dados no banco de dados sejam precisos e preservados contra atualizações válidas.

9. BANCA: CESPE ANO: 2015 ÓRGÃO: TRE-GO PROVA: TÉCNICO DO JUDICIÁRIO - PROGRAMAÇÃO DE SISTEMAS

Julgue os seguintes itens, a respeito da modelagem de dados.

[65] Considere a seguinte situação hipotética. Em um banco de dados referente a um curso, um aluno pode estar em mais de um curso ao mesmo tempo. Além disso, na tabela de cursos realizados por aluno, estão presentes as chaves estrangeiras aluno e curso. Nessa situação, tanto o código do curso como o código do aluno são chaves primárias nas tabelas curso e aluno, respectivamente.

[66] Ao se excluir uma tupla de um banco de dados, pode-se violar a integridade referencial desse banco por uma chave primária.

[67] Um conjunto de entidades que não possuem atributos suficientes para formar uma chave primária é definido como um conjunto de entidades fortes.

[68] Uma chave primária identifica um único valor de uma tupla no banco de dados e não possui mais de um atributo na tabela.

10. Ano: 2008 Banca: CESPE Órgão: STF Prova: Analista Judiciário - Tecnologia da Informação

O armazenamento e a recuperação de grandes quantidades de dados é um trabalho importante e muito explorado em um sistema gerenciador de banco de dados (SGBD). Com relação aos conceitos que envolvem esse sistema, julgue os itens que se seguem.

Integridade referencial pode ser definida como uma condição imposta a um conjunto de atributos de uma relação para que valores que apareçam nesse conjunto também apareçam em um certo conjunto de atributos de uma outra relação.

11. Ano: 2014 Banca: CESPE Órgão: SUFRAMA Prova: Analista Técnico - Tecnologia da Informação

Com relação aos sistemas gerenciadores de banco de dados (SGBD), julgue os itens a seguir.

A integridade semântica de um SGBD garante que os dados estejam sempre corretos em relação ao domínio de aplicação.



12. Ano: 2015 Banca: CESPE Órgão: TJDFT Prova: Programação de Sistemas

Julgue os itens seguintes a respeito de banco de dados.

[61] Em uma tabela de um banco de dados relacional, se uma restrição de chave primária for definida como composta de mais de uma coluna, os seus valores poderão ser duplicados em uma coluna; no entanto, cada combinação de valores de todas as colunas na definição da restrição de chave primária deve ser exclusiva.

13. Ano: 2018 Banca: CESPE Órgão: STJ Cargo: Técnico Judiciário – Suporte Técnico

Acerca de banco de dados, julgue os itens que se seguem.

73 Relacionamentos do tipo um-para-um podem ser representados em até três tabelas, de acordo com a obrigatoriedade do relacionamento.

77 Na criação de uma tabela para os clientes de uma organização, os atributos de CPF e CNPJ, para pessoas físicas e jurídicas, respectivamente, são a escolha mais indicada para representar a chave primária (PK) da tabela.

14. Ano: 2018 Banca: CESPE Órgão: ABIN Cargo: Área 08 Questão: 142

A respeito de sistemas gerenciadores de banco de dados, julgue os próximos itens.

142 Chave primária é o conjunto de um ou mais atributos para identificar uma tupla de uma entidade.

15. Ano: 2018 Banca: CESPE Órgão: STM Cargo: Programação de Sistemas Questão: 61 a 65

Acerca dos conceitos de normalização de dados e dos modelos de dados, julgue os itens subsequentes.

61 Uma tabela estará na segunda forma normal (2FN) quando, além de estar na terceira forma normal (3FN), ela contiver dependências funcionais parciais.

62 A passagem à terceira forma normal (3FN) tem como objetivo principal gerar o modelo lógico de dados; por isso, ela não visa eliminar redundância de dados, como ocorre com as demais formas normais.

65 A transformação do esquema de tabela não normalizada em um esquema relacional na primeira forma normal (1FN) consiste da eliminação das tabelas aninhadas.

16. BANCA: CESPE ANO: 2015 ÓRGÃO: CGE-PI PROVA: AUDITOR GOVERNAMENTAL - TECNOLOGIA DA INFORMAÇÃO

A respeito de banco de dados, julgue os itens subsequentes.

Para normalizar, conforme primeira forma, uma tabela em um banco de dados, é preciso criar chaves estrangeiras que representem a ligação entre elas.

17. BANCA: CESPE ANO: 2013 ÓRGÃO: CRPM PROVA: ANALISTA EM GEOCIÊNCIAS - SISTEMAS DE INFORMAÇÃO



No que concerne a mapeamento de dados lógico e físico e a elaboração e implantação de projeto de banco de dados, julgue os seguintes itens.

No processo de implantação de um projeto de banco de dados, devem ser utilizadas as operações de álgebra relacional de dados para estabelecer as restrições de cardinalidade e relacionamento entre o conjunto de entidades.

18. BANCA: CESPE ANO: 2013 ÓRGÃO: ANTT PROVA: ANALISTA ADMINISTRATIVO - DESENVOLVIMENTO DE SISTEMAS

Julgue os itens subsequentes, relativos a banco de dados.

A linguagem padrão de consulta SQL (structured query language) utiliza uma combinação de construtores em álgebra e cálculo relacional.

19. BANCA: CESPE ANO: 2013 ÓRGÃO: TCE-ES PROVA: ANALISTA ADMINISTRATIVO - INFORMÁTICA

O conjunto de operações cujo resultado seja uma nova relação e que envolve seleção, projeção, união e produto cartesiano é denominado

A mapeamento de cardinalidades.

B álgebra relacional.

C generalização.

D chave primária.

E herança.

20. CESPE - Auditor de Controle Externo (TCE-PA)/Informática/Analista de Sistema/2016

Julgue o item que se segue, relativo a modelagem de dados.

Dois diagramas de entidade de relacionamento são equivalentes se possuem entidades e relacionamentos que geram o mesmo esquema de banco de dados.

21. Ano: 2016 Banca: CESPE Órgão: TCE-SC Cargo: Auditor de TI

Com relação aos bancos de dados relacionais, julgue os próximos itens.

94 O catálogo de um sistema de gerenciamento de banco de dados relacional armazena a descrição da estrutura do banco de dados e contém informações a respeito de cada arquivo, do tipo e formato de armazenamento de cada item de dado e das restrições relativas aos dados.

95 Denomina-se visão uma tabela única derivada de uma ou mais tabelas básicas do banco. Essa tabela existe em forma física e viabiliza operações ilimitadas de atualização e consulta.

96 Em bancos de dados relacionais, as tabelas que compartilham um elemento de dado em comum podem ser combinadas para apresentar dados solicitados pelos usuários.

22. Ano: 2015 Banca: CESPE Órgão: TRE-PI – Questão 56



Acerca da aplicação dos princípios de normalização (Formas Normais), assinale a opção correta.

A A aplicação da 1FN se dá se e somente se, para todo modelo, for aplicada a Forma Normal de Boyce-Codd (ou BCNF).

B A 2FN é baseada no conceito de dependência funcional total, isto é, todo atributo não primário de uma entidade tem dependência funcional total da chave primária.

C A Terceira Forma Normal (3FN) requer que não haja dependências intransitivas de atributos que não sejam com toda chave candidata.

D A aplicação da Primeira Forma Normal (1FN) requer que, ao fim da sua aplicação, todos os atributos de uma relação sejam multivalorados ou estejam em tabelas aninhadas, o que garante grupos repetidos de dados, reduzindo o tamanho físico do banco de dados.

E A Segunda Forma Normal (2FN) requer que, ao fim da sua aplicação, não haja dependências transitivas de atributos que não sejam com toda chave candidata.

23. BANCA: CESPE ANO: 2015 ÓRGÃO: MPOG PROVA: ANALISTA - ANALISTA EM TECNOLOGIA DA INFORMAÇÃO

A respeito de modelo entidade-relacionamento e normalização, julgue os itens subsequentes.

113 Em relações normalizadas, na primeira forma normal, toda tupla em toda relação contém apenas um único valor, do tipo apropriado, em cada posição de atributo.

114 Sabendo que, nos relacionamentos ternários, a cardinalidade refere-se a pares de entidades, em um relacionamento ternário R entre três entidades A, B e C, a cardinalidade máxima de A e B dentro de R indica quantas ocorrências de C podem estar associadas a um par de ocorrências de A e B.

24. BANCA: CESPE ANO: 201 ÓRGÃO: CGE-PI PROVA: AUDITOR GOVERNAMENTAL - TECNOLOGIA DA INFORMAÇÃO

A respeito de banco de dados, julgue os itens subsequentes.

96 Um modelo de dados pode ser usado para representar os tipos de dados existentes em um banco de dados de um sistema online de reservas.

97 Em banco de dados relacional, os atributos representam as entidades do mundo real.

98 Em um relacionamento de tabelas de um banco de dados relacional, a chave estrangeira serve para referenciar uma entidade dentro de outra tabela, facilitando, assim, a busca e o agrupamento dessas entidades.

99 Para normalizar, conforme primeira forma, uma tabela em um banco de dados, é preciso criar chaves estrangeiras que representem a ligação entre elas.



100 Em um sistema gerenciador de banco de dados, a linguagem de definição de dados possibilita a criação das tabelas bem como a autorização de acesso aos dados para determinados usuários do banco de dados.



GABARITO

1. C(com ressalvas) C E E
2. E
3. E
4. C
5. C
6. D
7. C E E E
8. E
9. E
10. C
11. C
12. C
13. C E
14. C
15. E E C
16. E
17. E
18. C
19. B
20. C



CONSIDERAÇÕES FINAIS

Chegamos ao final da nossa aula sobre o modelo relacional! Considero esse um dos **assuntos mais densos do nosso curso**. Tente esgotar as questões sobre o assunto e, se possível, assistir as videoaulas com os conceitos associados ao tema.

A continuação desse assunto encontra-se na próxima aula. Devemos falar sobre a linguagem SQL, por meio do SQL é possível usar os conceitos do modelo relacional em Sistemas Gerenciadores de banco de dados para criar estruturas de armazenamento e manipular os dados sobre elas. Espero reencontrar você.

As videoaulas referentes a esse assunto já estão disponibilizadas no site do Estratégia Concursos®, não deixe de assistir e mandar suas observações!

Até a próxima!

Thiago Cavalcanti



ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



1 Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



2 Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



3 Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



4 Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



5 Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



6 Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



7 Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



8 O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.